Multiple Choice Questions

Subject: CP(3310701)	Semester: 1st

1. Who invented C Language.?
   a. Charles Babbage
   b. Grahambel
   c. Dennis Ritchie
   d. Steve Jobs

   Explanation:

   Full name is Dennis MacAlistair Ritchie. He also invented Unix Operating System along with his colleague Ken Thomson.

2. C is a which level language.?
   a. Low Level
   b. High Level
   c. Low + High
   d. None

3. C Language is a successor to which language.?
   a. FORTRAN
   b. D Language
   c. BASIC
   d. B Language

4. Low level language is .?
   a. Human readable like language.
   b. language with big program size.
   c. language with small program size.
   d. Difficult to understand and readability is questionable.

5. High level language is a .?
   a. Human readable like language.
   b. language with small program size.
   c. language with big program size.
   d. language which is difficult to understand and not human readable.

6. Which program outputs "Hello World.." .?
   a.
```
main()
{
 scanf("Hello World..");
}
```
   b.
```
main()
{
 printf("Hello World..");
}
```
   c.
```
main()
{
 print("Hello World..");
}
```
   d.
```
main()
{
 scan("Hello World..");
}
```

7. C is _____ type of programming language.?
   a. Object Oriented

b. Procedural

c. Bit level language

d. Functional

Explanation:

C is a procedural language. It is written in a number of steps using statements and functions. Logic is clearly depicted in the program. Procedural language is aslo called Imperative Language. Examples are COBAL, BASIC etc.

8. What is the present C Language Standard.?
   a. C99 ISO/IEC 9899:1999
   b. C11 ISO/IEC 9899:2011
   c. C05 ISO/IEC 9899:2005
   d. C10 ISO/IEC 9899:2010

9. What are the new features of C11 or ISO IEC 9899 2011 standard.?
   a. Type generic Macros, Static Assertions
   b. Multi Threading, Anonymous Unions, quick_exit
   c. Bounds Checking Interfaces, Anonymous Strurctures
   d. All

10. C language was invented in which laboratories.?
    a. Uniliver Labs
    b. IBM Labs
    c. AT&T Bell Labs
    d. Verizon Labs

    Explanation:

    C was invented in Bell Laboratories in New Jersey.

11. BCPL Language is also called..?
    a. C Language
    b. B Language
    c. D Language
    d. None

    Explanation:

    B language is successor of BCPL (Basic Combned Programming Language). B language was invented by Ken Thomson.

12. C language was invented to develop which Operating System.?
    a. Android
    b. Linux
    c. Ubuntu
    d. Unix

    Explanation:

    C was invented to develop Unix Operating System to overcome compatibility with different Hardware Platforms.

13. C language was invented in the year.?
    a. 1999
    b. 1978
    c. 1972
    d. 1990

14. C language is used in the development of .?
    a. Databases
    b. Graphic applications
    c. Word Processors
    d. All of the above

Explanation:

C language is very efficient in using hardware resources.

15. A C program is a combination of.?
    a. Statements
    b. Functions
    c. Variables
    d. All of the above

16. Choose correct answer..

```
#include<stdio.h>
main()
{
/* Multi Line Comment
This line is ignored by compiler
*/
printf("Hello C..");
}
```

    a. #include is a Preprocessor Directive
    b. <stdio.h> is a header file with predefined functions like printf, scanf etc
    c. #include
       main()
       {
       }
       is a mandatory function to be included in every C Program.
    d. All the above

17. Correct way of commenting a single line is.?

a. /*printf("Hello C..");
printf("How are you.");

b. //printf("Hello C..");
printf("How are you.");

c. /*printf("Hello C..");
printf("How are you.");*/

d. /printf("Hello C..");/
printf("How are you.");

Explanation:

Answer C comments two lines with Multi-Line comment or BLOCK Comment characters /* ....*/

Only // is a Single Line Commenting characters.

18. Single Line Comment // is also called.?

a. C++ Style Comment
b. Java Style Comment
c. PHP Style Comment
d. All the above

19. What is an Identifier in C Language.?

a. Name of a Function or Variable
b. Name of a Macros
c. Name of Structure or Union
d. All the above.

Explanation: int age=25; //here age is an Identifier

20. An Identifier may contain?
    a. Letters a-z, A-Z in Basic character set. Unicode alphabet characters other languages
    b. Underscore _ symbol
    c. Numbers 0 to 9 Unicode Numbers in other languages
    d. All the above

21. What is the number of characters used to distinguish Identifier or Names of Functions and Global variables.?
    a. 31
    b. 32
    c. 33
    d. 28

Explanation:

First 31 characters in general. If first 31 characters are same for two different identifiers, compiler gets confused.

22. What is length of an Identifier that is unique for Non Global Variables and Non Function Names.?
    a. 32
    b. 63
    c. 64
    d. 68

Explanation: if 31 is present choose. Because old compilers support up to 31 only.

Upto first 63 characters you can show differentiation in the name of say

int abcdefghijklmnopqrstuvwxyz1234567788= 10;

int abcdefghijklmnopqrstuvwxyz1234567799 = 20;

23. An Identifier can start with.?

a. Alphabet

b. Underscore (_) sign

c. Any character that can be typed on a keyboard

d. Option A & Option B

Explanation: Identifier is just a name given to a Function, Variable etc.

Identifier name should contain only Letter, Numbers and Underscore.

24. C Programs are used in .?

a. Any Electronic device which works on some logic and Operating System.

b. Washing machine

c. Fridge, Microwave Ovens

d. All the above.

Explanation:

C is very fast to execute and safe to embed along with microprocessors. Device drivers are written in C and C++.

25. What are the types of Constants in C Language.?

a. Primary Constants

b. Secondary Constants

c. Basic Constants and Advanced Constants

d. Primary Constants and Secondary Constants

Explanation:

Primary Constants are Integer (int), Floating Point (float) , Character (char)

Secondary Constants are Structure, Union, Array and Enum.

26. Choose correct statements

a. A constant value does not change. A variable value can change according to needs.

b. A constant can change its values. A variable can have one constant value only.

c. There is no restriction on number of values for constants or variables.

d. Constants and Variables can not be used in a singe main function.

Explanation:

Constant value is always constant. Constant is also called Literal.

Variable can have any number of arbitrary values and once value at any point of time. Variable is also called Identifier.

27. Find an integer constant.
    a. 3.145
    b. 34
    c. "125"
    d. None of the above

Explanation:

Integer constant is a full or whole number without any decimal point. So 3.14 is a floating point number or Real number.

28. Find a Floating Point constant.
    a. 12.3E5
    b. 12e34
    c. 125.34857
    d. All the above.

Explanation:

Floating Point can be represented in two forms.

1. Fractional Form

eg. 12345.67

2. Exponential Form

(Mantissa)e(number) or (Mantissa)E(number)

eg. 123.4567E2

(e2 = 10 power 2 = 100)

29. Find a Character constant.
    a. A'

       'a'

    b.

       '1'

       '9'

    c.

       '$'

       '#'

    d. All the above.

Explanation:

A character constant contains only one character within Single Quotes. ' '. Single Quote is typed using Single Quote Double Quote Key near Enter Key in a Keyboard. Simply it is Right Single Quote.

Left Single Quote looks like this `

Right Single Quote looks like this '

30. A Variable of a particular type can hold only a constant of the same type. Choose right answer.
    a. TRUE
    b. FALSE
    c. It depends on the place the variable is declared.
    d. None of the above.

Explanation:

An int can hold only Integer constant.

A float can hold only Real Number constants.

A char can hold only Character constants.

31. Choose a right statement.
    a. int myage = 10;

       int my_age = 10;
    b. int myage = 10;

       int my,age = 10;
    c. int myage = 10;

       int my age = 10;
    d. All are right

Explanation:

Only Underscore (_) symbol is allowed in a variable name i.e identifier name. Space, Comma and other special characters are not allowed.

32. Number of Keywords present in C Language are .?
    a. 32
    b. 34
    c. 62
    d. 64

Explanation:

Only 32 Keywords originally. Compilers are individual companies can include and use extra keywords if required. Such keywords should preceed with __ (two Underscore symbols before names).

eg. __mykeyword

33. Each statement in a C program should end with.?

a. Semicolon ;
b. Colon :
c. Period . (dot symbol)
d. None of the above.

Explanation:

e.g

int amount = 10;

float a,b;

34. Choose a correct statement.
a. Compiler converts your C program into machine readable language.
b. C Editor allows you to type C Programs. It is just like a Notepad with extra options.
c. Console shows the output of a C Program if it is text output.
d. All the above

35. Identify wrong C Keywords below.
a. auto, double, int, struct
b. break, else, long, switch
c. case, enum, register, typedef
d. char, extern, intern, return

Explanation:

'intern' is not a keyword. Remaining are all valid keywords.

36. Identify wrong C Keywords below.
a. union, const, var, float

b. short, unsigned, continue, for

c. signed, void, default, goto

d. sizeof, volatile, do, if

Explanation:

'var' is not a valid keyword.

37. Identify wrong C Keywords below.

a. static, while, break, goto

b. struct, construct, signed, unsigned

c. short, long, if, else

d. return, enum, struct, do

Explanation:

construct is not a keyword.

All 32 Keywords are given for reference. auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while.

38. Find a correct C Keyword below.

a. breaker

b. go to

c. shorter

d. default

39. Find a correct C Keyword below.

a. work

b. case

c. constant

d. permanent

40. Find a correct C Keyword.

a. Float

b. Int

c. Long

d. double

Explanation:

All C Keywords are in lower case.

41. Types of Integers are.?

a. short

b. int

c. long

d. All the above

Explanation:

Size of int < long.

42. Types of Real numbers in C are.?

a. float

b. double

c. long double

d. All the above

Explanation:

Size of float < double < long double

43. signed and unsigned representation is available for.?
    a. short, int, long, char
    b. float, double, long double
    c. A & B
    d. None of the above

Explanation:

Real numbers like float, double and long double do not support unsigned representation.

44. Size of a Turbo C C++ compiler is.?
    a. 16 bit
    b. 32 bit
    c. 64 bit
    d. 128 bit

45. Sizes of short, int and long in a Turbo C C++ compiler in bytes are.?
    a. 2, 2, 4
    b. 2, 4, 4
    c. 4, 8, 16
    d. 8, 8, 16

46. Sizes of short, int and long in Visual Studio or GCC compiler in bytes are.?
    a. 2, 2, 4
    b. 2, 4, 4
    c. 4, 4, 8
    d. 4, 8, 8

47. Range of signed char and unsigned char are.?

a. -128 to +127

   0 to 255

b. 0 to 255

   -128 to +127

c. -128 to -1

   0 to +127

d. 0 to +127

   -128 to -1

Explanation:

Advantage of an unsigned representation is only to increase the upper limit i.e positive limit. Size of a char remains same i.e 1 Byte.

48. Ranges of signed int and unsigned int are.?

a. 0 to 65535

   -32768 to +32767

b. -32768 to +32767

   0 to 65535

c. -32767 to +32768

   0 to 65536

d. 0 to 65536

   -32767 to +32768

Explanation:

Default assumption is Turbo C/C++, 16 bit compiler. Size of an int is 2 bytes for both signed and unsigned representation.

49. Size of float, double and long double in Bytes are.?

a. 4, 8, 16

b. 4, 8, 10

c. 2, 4, 6

d. 4, 6, 8

Explanation:

Real numbers are represented in float, double and long double format.

eg. float interest = 12.55f;

50. Range of singed long and unsigned long variables are.?

a. -2147483647 to +2147483648

0 to 4294967295

b. -2147483648 to +2147483647

0 to 4294967296

c. -2147483648 to +2147483647

0 to 4294967295

d. 0 to 4294967295

-2147483648 to +2147483647

Explanation:

Size of a long variable is 4 Bytes or 32 bits. (2)^32

51. Range of float variable is.?

a. -3.2e38 to +3.2e38

b. -3.8e32 to +3.8e32

c. -3.4e34 to +3.4e34

d. -3.4e38 to +3.4e38

Explanation:

e represents exponential.

52. Left most bit 0 in Singed representation indicates.?

a. A Positive number

b. A Negative Number

c. An Unsigned number

d. None of the above

Explanation:

For negative numbers 1 is used as a left most bit.

53. If you do not specify a storage class for a Variable.?

a. You get compiler error.

b. You get a compiler warning.

c. Output is null always

d. None of the above

Explanation:

Yes. Even if you do not specify a Storage class for a Variable, AUTOMATIC storage class is applied.

54. What is a C Storage Class.?

a. C Storage decides where to or which memory store the variable.

b. C Storage Class decides what is the default value of a variable.

c. C Storage Class decides what is the Scope and Life of a variable.

d. All the above.

55. Every C Variable must have.?

a. Type

b. Storage Class

c. Both Type and Storage Class

d. Either Type or Storage Class

Explanation:

Yes. A C Variable should have both Type and Storage Class.

Eg.

int a=5; //By default its Storage Class is auto.

auto int b=10;

56. Find a C Storage Class below.
    a. static
    b. auto
    c. register & extern
    d. All the above

57. What is the default C Storage Class for a variable.?
    a. static
    b. auto
    c. register
    d. extern

Explanation:

int a=5; // auto is by default

auto int b=10; // storage class auto is explicitly mentioned.

58. Choose a right answer.
    a. auto variable is stored in 'Memory'.
       static variable is stored in 'Memory'.
       extern variable is stored in 'Memory'.
       register variable is stored in 'Memory'.
    b. auto variable is stored in 'Memory'.
       static variable is stored in 'Memory'.
       extern variable is stored in 'Memory'.
       register variable is stored in 'Register'.

c. auto variable is stored in 'Register'.
static variable is stored in 'Register'.
extern variable is stored in 'Register'.
register variable is stored in 'Memory'.

d. auto variable is stored in 'Register'.
static variable is stored in 'Register'.
extern variable is stored in 'Register'.
register variable is stored in 'Register'.

Explanation:

Only a register variable is stored in CPU Memory called Registers. Other variables are stored in RAM.

59. A register variable is stored in a Register. Where does a Register Present in a Computer.?

a. RAM (Random Access Memory)
b. ROM (Read Only Memory)
c. CPU (Central Processing Unit)
d. DMA (Direct Memory Access)

Explanation:

Yes. Registers are part of a CPU to store variable whose value changes frequently. Loop variables for example.

```
register int i=1;

while(i <= 10)

{

        printf("i=%d\n", i);

}
```

60. Variables of type auto, static and extern are all stored in .?
    a. ROM
    b. RAM
    c. CPU
    d. Compiler

61. Which among the following is a Local Variable.?
    a. register
    b. auto
    c. static
    d. extern

62. Which among the following is a Global Variable.?
    a. auto
    b. register
    c. static
    d. extern

63. Choose a correct statement about static variable.
    a. A static global variable can be accessed in other files.
    b. A static global variable can be used only in a file in which it is declared.
    c. A static global variable can not be declared without extern keyword.
    d. Default value of a static variable is -1.

64. register float a = 3.14f; Choose right statement.
    a. Variable a is stored in CPU registers for fast access.
    b. Variable a is converted to int and then stored in a CPU register.

c. register Storage Class is ignored and treated as

auto float a = 3.14f;

d. You get a compiler error as you can not store non integer value in a CPU register.

Explanation:

Yes. CPU register can not store anything more than 16 bits or 2 bytes. You will not any compiler errors. It will be treated just like an auto Storage Class variable.

65. What is the difference between Declaration and Definition.?

a. Declaration does allocate memory for a variable.

Definition does allocate memory for a variable.

b. Declaration does allocate memory for a variable.

Definition does not allocate memory for a variable.

c. Declaration does not allocate memory for a variable.

Definition does allocate memory for a variable.

d. Declaration does not allocate memory for a variable.

Definition does not allocate memory for a variable.

Explanation:

int a; //Definition. Allocates memory.

extern int b; //Does not allocate memory.

66. Choose a right statement.

a. A non static global variable can not be used in included files.

b. A non static global variable can be used or referred to inside included files.

c. A non static global variable does not live till the end of program execution.

d. None of the above

67. Choose a right statement.

a. Redeclaration of a variable is Ok.

b. Redefinition of a variable is not Ok.

c. Definition of a variable uses memory blocks.

d. All the above.

68. Choose a correct statement.

a. Register variables are usually fast retrieving variables.

b. Static variables are usually maintain their values between function calls.

c. Auto variables release their memory after the block or function where they are declared.

d. All the above.

69. Choose a right statement.

a. Variables of type auto are stored in Stack memory.

b. Variable of type Static are stored in Segmented Memory.

c. Variables of type register are stored in Micro Processor Memory.

d. All the above.

70. Choose a right statement.

a. Variables of type auto are initialized fresh for each block or function call.

b. Variables of type static are initialized only first time the block or function is called.

c. Variables of type register are initialized each time the block or function is executed.

d. All the above.

71. Choose a correct statement regarding automatic variables.

a. #include<stdio.h>

main()

```
{
        auto int a;
        printf("%d", a);
}
//output is compiler error. a is not initialized.
```

b.
```
#include<stdio.h>
main()
{
        auto int a;
        printf("%d", a);
}
//output = 0
```

c.
```
#include<stdio.h>
main()
{
        auto int a;
        printf("%d", a);
}
//output = null
```

d.
```
#include<stdio.h>
main()
{
        auto int a;
        printf("%d", a);
}
//output = some random number
```

Explanation:

Yes. If an integer is not initialized, some random number in integer range will be displayed.

72. What is the output of the program.?

```
#include<stdio.h>
        int main()
       {
            printf("Hello Boss.");
        }
```

a. Hello Boss.
b. hello boss
c. No output
d. Compiler error

Explanation:

Notice the return type int before main() method. So main should maintain a return 0; or return somenumber; statement.

73. What is the output of the program.?

```
#include<stdio.h>
static int k;
void main()
{
        printf("%d", k);
}
```

a. -1

b. 0

c. 90

d. Compiler error

Explanation:

Default value of a static variable is zero by default.

74. Choose a correct statement.

int a = 12 + 3 * 5 / 4 – 10

a. 12, 3, 5, 4 and 10 are Operators.
+, -, * and / are Operands.
= is an increment operator.

b. 12, 3, 5, 4 and 10 are Operands.
+, -, * and / are Operators.
= is decrement operator.

c. 12, 3, 5, 4 and 10 are Operands.
+, -, * and / are Operators.
= is an assignment operator.

d. 12, 3, 5, 4 and 10 are Operands.
+, -, * and / are Logical Operators.
= is an assignment operator.

75. Operator % in C Language is called.?

a. Percentage Operator

b. Quotient Operator

c. Modulus

d. Division

Explanation:

Operator % is called Modulus or Modular or Modulo Division operator in C. It gives the reminder of the division.

int a = 11%4;

Now a holds only 3 which is the reminder.

76. Output of an arithmetic expression with integers and real numbers is ___ by default.?

a. Integer

b. Real number

c. Depends on the numbers used in the expression.

d. None of the above

Explanation:

Any arithmetic operation with both integers and real numbers yield output as Real number only.

5 + 10.56 = 15.560000 which is a real number.

5 + 10.0 = 15.000000 is also a real number.

77. Choose a right statement.

int a = 10 + 4.867;

a. a = 10

b. a = 14.867

c. a = 14

d. compiler error.

Explanation: a is an int variable. So 10+4.867 = 14.867 is truncated to 14 and assigned to a.

78. Choose a right statement.

```
int a = 3.5 + 4.5;
```

a. a = 0
b. a = 7
c. a = 8
d. a = 8.0

Explanation:

3.5 + 4.5 = 8.0 is a real number. So it is converted to downgraded to int value. So a = 8.

79. Choose a right statement.

```
float var = 3.5 + 4.5;
```

a. var = 8.0
b. var = 8
c. var = 7
d. var = 0.0

Explanation:

A float variable can hold a real number.

80. Choose right statement.

```
int main()
{
   float c = 3.5 + 4.5;
   printf("%f", c);
   return 0;
}
```

a. 8.0

b. 8.000000

c. 8

d. 7

Explanation:

Float can print precision up to 6 digits. So 6 zeros will be shown if there are no digits after decimal point.

81. Choose a right statement.

```
void main()
{
   float c = 3.5 + 4.5;
   printf("%d", (int)c);
}
```

a. 8.0

b. 8.000000

c. 7

d. 8

Explanation:

You are printing a float variable by type casting to int. So integer is printed.

int c = 3.5 + 4.5 also holds and prints 8.

82. Choose a right statement.

```
int a = 5/2;
int b = 5.0/2;
```

int c = 5 / 2.0;

int d = 5.0/2.0;

a. a = 2, b = 2, c = 2, d= 2
b. a = 2, b = 2.0, c = 2, d= 2.0
c. a = 2, b = 2.5, c = 2.5, d= 2.5
d. a = 2.5, b = 2.5, c = 2.5, d= 2.5

Explanation:

Irrespective of numbers after decimal point, an int variable holds only integer value i.e 2.

83. Choose a right statement.

float a = 5/2;

float b = 5/2.0;

float c = 5.0/2;

float d = 5.0/2.0;

a. a=2.5, b=2.5, c=2.5, d=2.5
b. a=2, b=2.5, c=2.5, d=2.5
c. a=2.0, b=2.5, c=2.5, d=2.5
d. a=2.0, b=2.0, c=2.0, d=2.0

Explanation:

In division, to get the actual real value, you should specify at least one real number.

Variable a holds only 2. But variables b,c and d contain real numbers as either numerator or denominator is a real number.

84. Choose a right statement.

int var = 3.5;

a. a = 3.5

b. a = 3

c. a = 0

d. Compiler error

Explanation:

a stores only integer value. So, 3.5 is truncated to 3.

85. What is the priority of operators *, / and % in C language.?

a. * > / > %

b. % > * > /

c. Both % = / , * are same

d. All three operators *, / and % are same.

Explanation:

Operators Multiplication *, Division / and Modulo Division % are all having the same Priority.

86. In C language, which Operator group has more priority between (*, / and %) and (+, -) groups.?

a. Both groups share equal priority.

b. (+, -) > (*, / and %)

c. (+, -) < (*, / and %)

d. None of the above.

Explanation:

+ and - has same priority. *, / and % has equal priority. But (+, -) has less priority than (*, / and %).

87. What is the Priority among (*, /, %), (+, -) and (=) C Operators.?

a. (*, /, %) > (+, -) < (=)

b. (*, /, %) < (+, -) < (=)

c. (*, /, %) > (+, -) > (=)

d. (*, /, %) < (+, -)

(+, -) == (=)

Explanation:

Assignment operator in C has the least priority.

88. What is the output of the C statement.?

```
void main()
{
        int a=0;
        a = 4 + 4/2*5 + 20;
       printf("%d", a);
}
```

a. 40

b. 4

c. 34

d. 54

Explanation:

/ and * has equal priority. But associativity is from L to R.

4 + 2*5 + 20

4 + 10 + 20 = 34

89. What is the output of the C Program.?

```
void main()
{
        int a=0;
        a = 10 + 5 * 2 * 8 / 2 + 4;
       printf("%d", a);
}
```

a. 124

b. 54

c. 23

d. 404

Explanation:

10 + 10*8/2 + 4

10 + 80/2 + 4

10 + 40 + 4 = 54

90. Choose a C Conditional Operator from the list.

a. ?:

b. :?

c. :<

d. <:

Explanation:

?: = Question Mark Colon is also called C Ternary Operator.

91. What is the other name for C Language ?: Question Mark Colon Operator.?

a. Comparison Operator

b. If-Else Operator

c. Binary Operator

d. Ternary Operator

92. Choose a syntax for C Ternary Operator from the list.

a. condition ? expression1 : expression2

b. condition : expression1 ? expression2

c. condition ? expression1 < expression2

d. condition < expression1 ? expression2

Explanation:

If the condition is true, expression 1 is evaluated. If the condition is false, expression 2 is evaluated.

93. What is the output of the C statement.?

```
void main()
{
        int a=0;
        a = 5<2 ? 4 : 3;
       printf("%d",a);
}
```

a. 4

b. 3

c. 5

d. 2

Explanation:

5<2 is false. So 3 will be picked and assigned to the variable a.

94. Choose a correct statement regarding C Comparison Operators.

a. (x == y) Is x really equal to y.

   (x != y) Is x not equal to y.

b. (x < y) Is x less than y

   (x > y) Is x greater than y

c. (x <= y) Is x less than or equal to y.

   (x >= y) Is x greater than or equal to y

d. All the above

95. Choose a statement to use C If Else statement.

a. else if is compulsory to use with if statement.

b. else is compulsory to use with if statement.

c. else or else if is optional with if statement.

d. None of the above

96. Choose a correct C Statement using IF Conditional Statement.

a.
```
if( condition )
{
   //statements;
}
```

b.
```
if( condition )
{
   //statements;
}
else
{
   //statements;
}
```

c.
```
if( condition1 )
{
   //statements;
}
else if( condition2)
{
   //statements;
}
else
{
   //statements;
}
```

d. All the above.

97. What is the output of the C Program.?

```
void main()
{
   if( 4 > 5 )
   {
      printf("Hurray..\n");
   }
    printf("Yes");
}
```

a. Yes
b. Hurray..
   Yes
c. Hurray..Yes
d. Compiler error

Explanation:

if condition fails. So control will not enter Hurray printf statement.

98. What is the output of the C Program.?

```
void main()
{
   if( 4 > 5 )
      printf("Hurray..\n");
     printf("Yes");
}
```

a. Yes
b. Hurray..
   Yes
c. Hurray..Yes
d. No Output

Explanation:

To include more than one statement inside If block, use { } braces. Otherwise, only first statement after if block is included. IF condition fails with false. So second if which is outside of If is executed.

99. What is the output of the C Program.?

```
int main()
{
   if( 4 < 5 )
      printf("Hurray..\n");
      printf("Yes");
   else
      printf("England")
   return 0;
}
```

a. Hurray..Yes

b. Hurray..

   Yes

c. Compiler error

d. None of the above

Explanation:

If block includes only Single Hurray printf statement without curly braces { }. So second Yes printf statement is not part of IF block. Else should immediately follow IF block. Otherwise, compiler throws errors. To compile well, use { } braces for two printf statements or remove second printf after IF.

100. What is the output of the C Program.?

```
int main()
{
   if( 10 < 9 )
      printf("Hurray..\n");
```

```
    else if(4 > 2)
        printf("England");
        return 0;
}
```

a. England

b. Hurray..

c. Compiler error for missing else

d. None of the above

Explanation:

You can omit ELSE comfortably. Compiler will not complain above ELSE after IF or ELSE IF.

101. What is the output of C Program.?

```
int main()
{
    if( 10 > 9 )
        printf("Singapore\n");
    else if(4%2 == 0)
        printf("England\n");
        printf("Poland");
    return 0;
}
```

a. Singapore

b. Singapore
   Poland

c. Singapore
   England
   Poland

d. England
   Poland

Explanation:

Observe that Poland printf is not under ELSE IF as there are two statements without curly braces { }. IF condition is TRUE. So, ELSE IF will not be seen at all even though 4%2 == 0 is true.

102. What is the Priority of C Logical Operators.? NOT (!), AND (&&) and OR (||)

a. NOT (!) > AND (&&) > OR (||)

b. NOT (!) > AND (&&) = OR (||)

c. AND (&&) > OR (||) > NOT (!)

d. AND (&&) = OR (||) > NOT (!)

Explanation:

Logical NOT Operator in C has the highest priority.

103. What is the output of C Program.?

```
int main()
{
   int a=9, b;
   b = (a==9) ? (printf("CAT\n");printf("DOG\n")) : (printf("FOX"));
   return 0;
}
```

a. CAT

DOG

b. FOX

c. CAT

DOG

FOX

d. Compiler error

Explanation:

You can not put more than 1 statement inside expression1 or expression2 inside Ternary Operator.

(Condition)?(expression1):(expression2)

104. What is the output of C Program.?

```
int main()
{
   int a=9, b=6;
   if(a==9 && b==6)
   {
      printf("Hockey");
   }
   else
   {
      printf("Cricket");
   }
   return 0;
}
```

a. Cricket
   Football
b. Hockey
   Football
c. Football
d. Compiler error

Explanation:

== operator has more priority than &&. Logical && AND operator returns true only if both expressions are true.

105. What is the output of C Program.?

```
int main()
{
   int a=9, b=6;
   if(a!=9 || b==6)
   {
```

```
        printf("Hockey\n");
    }
    else
    {
        printf("Cricket\n");
    }

    printf("Football");
    return 0;
}
```

a. Cricket

   Football

b. Hockey

   Football

c. Football

d. Compiler error

Explanation:

Logical OR || operator returns true if any one expression is true.

106. Choose a correct C Operator Priority.? Items in one group ( ) has same priority.

a. (!) < (*, /, %) < (+, -) < (<, <=, >, >=) < (==, !=) < (&&) < (||) < (=)

b. ((!) , (*, /, %) , (+, -)) < (<, <=, >, >=) < (==, !=) < (&&) < (||) < (=)

c. (!) > (*, /, %) > (+, -) > (<, <=, >, >=) > (==, !=) > (&&) > (||) > (=)

d. ((!) , (*, /, %) , (+, -)) > (<, <=, >, >=) > (==, !=) > (&&) > (||) > (=)

Explanation:

(! Logical NOT) > (*, /, % Arithmetic) > (+, - Arithmetic) > (<, <=, >, >= Relational) > (==, != Relational) > (&& Logical AND) > (|| Logical OR) > (= Assignment).

107. Choose a right C Statement.

a. Loops or Repetition block executes a group of statements repeatedly.
b. Loop is usually executed as long as a condition is met.
c. Loops usually take advantage of Loop Counter
d. All the above.

108. Loops in C Language are implemented using.?
a. While Block
b. For Block
c. Do While Block
d. All the above

109. Which loop is faster in C Language, for, while or Do While.?
a. for
b. while
c. do while
d. All work at same speed

110. Choose correct C while loop syntax.
a.
```
while(condition)
{
  //statements
}
```
b.
```
{
  //statements
while(condition)
```
c.
```
 while(condition);
{
  //statements
}
```
d.
```
while()
{
  if(condition)
  {
    //statements
```

```
    }
}
```

111. Choose a correct C for loop syntax.

a. for(initalization; condition; incrementoperation)
```
{
  //statements
}
```
b. for(declaration; condition; incrementoperation)
```
{
  //statements
}
```
c. for(declaration; incrementoperation; condition)
```
{
  //statements
}
```
d. for(initalization; condition; incrementoperation;)
```
{
  //statements
}
```

Explanation:

increment or decrement operation at third place.

112. Choose a correct C do while syntax.

a. dowhile(condition)
```
{
  //statements
}
```
b. do while(condition)
```
{
  //statements
}
```
c. do
```
{
  //statements
}while(condition)
```
d. do

```
{
   //statements
}while(condition);
```

Explanation:

Semicolon after while(condition) is a must.

113. What is the output of C Program.?

```
int main()
{
   while(true)
   {
      printf("RABBIT");
      break;
   }
   return 0;
}
```

a. RABBIT

b. RABBIT is printed unlimited number of times.

c. No output

d. Compiler error.

Explanation:

while(TRUE) or while(true) does not work. true is not a keyword.

114. What is the output of C Program.?

```
int main()
{
   int a=5;
   while(a==5)
   {
      printf("RABBIT");
      break;
   }
   return 0;
```

```
}
```

a. RABBIT is printed unlimited number of times

b. RABBIT

c. Compiler error

d. None of the above.

Explanation:

If there is no BREAK statement, while loop runs continuously util the computer hangs. BREAK causes the loop to break once and the statement below the while if any will be executed.

115. What is the output of C Program.?

```
int main()
{
 int a=5;
   while(a=123)
   {
      printf("RABBIT\n");
      break;
   }
   printf("GREEN");
   return 0;
}
```

a. GREEN

b. RABBIT

GREEN

c. RABBIT is printed unlimited number of times.

d. Compiler error.

Explanation:

while(a=123) = while(123) = while(Non Zero Number). So while is executed. BREAK breaks the loop immediately. Without break statement, while loop runs infinite number of times.

116. What is the output of C Program.?

```
int main()
    {
       int a=25;
      while(a <= 27)
       {
          printf("%d ", a);
          a++;
       }
       return 0;
    }
```

a. 25 25 25

b. 25 26 27

c. 27 27 27

d. Compiler error

Explanation:

a++ is equivalent to a=a+1;

a is incremented each time.

117. What is the output of C Program.?

```
int main()
    {
       int a=32;
       do
       {
          printf("%d ", a);
          a++;
       }while(a <= 30);
       return 0;
    }
```

a. 32

b. 33

c. 30

d. No Output

Explanation:

do { } block is executed even before checking while(condition) at least once. This prints 32. To loop for the second time, while (32 <= 30) fails. So, loop is quit.

118. What is the output of C Program.?

```
int main()
{
   int a=32;
   do
   {
      printf("%d ", a);
      a++;
      if(a > 35)
         break;
   }while(1);
   return 0;
}
```

a. No Output

b. 32 33 34

c. 32 33 34 35

d. Compiler error

Explanation:

while(1) is infinite loop. So we kept if(condition) to break the loop. a++ is equivalent to a=a+1;

119. Choose a correct C Statement.

a. a++ is (a=a+1) POST INCREMENT Operator

b. a-- is (a=a-1) POST DECREMENT Opeartor

--a is (a=a-1) PRE DECREMENT Opeator

c. ++a is (a=a+1) PRE INCRMENT Operator

d. All the above.

120. Choose correct Syntax for C Arithmetic Compound Assignment Operators.

a. a+=b is (a= a+ b)

   a-=b is (a= a-b)

b. a*=b is (a=a*b)

   a/=b is (a = a/b)

c. a%=b is (a=a%b)

d. All the above.

121. What is the output of C Program.?

```
int main()
  {
    int k, j;
    for(k=1, j=10; k <= 5; k++)
    {
       printf("%d ", (k+j));
    }
    return 0;
  }
```

a. compiler error

b. 10 10 10 10 10

c. 10 11 12 13 14 15

d. None of the above

Explanation:

You can initialize any number of variables inside for loop.

122. What is the output of C Program.?

```
int main()
  {
    int k;
    for(k=1; k <= 5; k++);
    {
       printf("%d ", k);
    }
```

```
    return 0;
}
```

a. 1 2 3 4 5

b. 1 2 3 4

c. 6

d. 5

Explanation:

Semicolon at the end of for(); isolates the below print() block. After for loop is over, k value is 6.

```
for(k=1; k <= 5; k++)
{
   ;
}
{
   printf("%d ", k);
}
```

123. What is the output of C Program.?

```
int main()
{
  int k;
   for(;;)
   {
      printf("TESTING\n");
      break;
   }
   return 0;
}
```

a. No Output

b. TESTING

c. Compiler error

d. None of the above

Explanation:

for(;;) loop need not contain any initialization, condition and incre/decrement sections. All are optional. BREAK breaks the FOR Loop.

124. What is the output of C Program.?

```
int main()
  {
    int k;
    for(printf("FLOWER "); printf("YELLOW "); printf("FRUITS "))
    {
       break;
    }
    return 0;
  }
```

a. Compiler error
b. FLOWER FRUITS
c. FLOWER YELLOW
d. FLOWER YELLOW FRUITS

Explanation:

for(anything; anything; anything) is Ok. printf("YELLOW") prints YELLOW and returns 1 as result. So for loop runs forever. Actually break is saving us from quitting the for loop. Only after checking condition and executing the loop statements, third section is executed. break causes the loop to quit without incre/decrement section.

125. What is the way to suddenly come out of or Quit any Loop in C Language.?

a. continue; statement
b. break; statement
c. leave; statement
d. quit; statement

Explanation:

eg.

```
while(condition)
{
break;
```

```
}
```

126. Choose facts about continue; statement is C Language.

a. continue; is used to take the execution control to next iteration or sequence

b. continue; statement causes the statements below it to skip for execution

c. continue; is usually accompanied by IF statement.

d. All the above.

127. What is the output of C Program.?

```
int main()
{
   int a=14;
   while(a<20)
   {
      ++a;
      if(a>=16 && a<=18)
      {
         continue;
      }
      printf("%d ", a);

   }
   return 0;
}
```

a. 15 16 17 18 19

b. 15 18 19

c. 15 16 20

d. 15 19 20

Explanation:

Between 16 - 18, continue statement skips all other statements below it. So a will not be printed during that time.

15 printed

16 not printed

17 not printed

18 not printed

19 printed

20 printed

128. Choose a correct statement about C break; statement.?
   a. break; statement can be used inside switch block
   b. break; statement can be used with loops like for, while and do while.
   c. break; statement causes only the same or inner loop where break; is present to quit suddenly.
   d. All the above.

129. Choose a correct statement about C language break; statement.
   a. A single break; statement can force execution control to come out of only one loop.
   b. A single break; statement can force execution control to come out of a maximum of two nested loops.
   c. A single break; statement can force execution control to come out of a maximum of three nested loops.
   d. None of the above.

130. Choose a correct C Statement regarding for loop.

for(; ;);

   a. for loop works exactly first time
   b. for loop works infinite number of times
   c. Compiler error
   d. None of the above

Explanation: We are not specifying condition to exit the loop. Eg. for(a=0;a<10;a++)

131. What is the output of C Program.?

```
int main()
{
   int a=10, b, c;
   b=a++;
   c=++a;
   printf("%d %d %d", a, b, c);
   return 0;
}
```

a. 10 11 12

b. 12 10 12

c. 12 11 12

d. 12 12 12

Explanation:

a++ first assigns 10 to b. Next a is incremented separately. ++a increments from 11 to 12. Final ++a value is assigned to the left side variable C.

132. What is the output of C Program.?

```
int main()
{
   int a=0, b=0;
   while(++a < 4)
      printf("%d ", a);
   while(b++ < 4)
      printf("%d ", b);
   return 0;
}
```

a. 0 1 2 3 1 2 3 4

b. 1 2 3 1 2 3 4

c. 1 2 3 4 1 2 3 4

d. 1 2 3 4 0 1 2 3

Explanation:

(++a < 4) first increments and compares afterwards. (b++ < 4) first compares and increments afterwards.

133. What are C ASCII character ranges.?

a. A to Z = 65 to 91
b. a to z = 97 to 122
c. 0 to 9 = 48 to 57
d. All the above

Explanation:

All remaining characters are special characters or symbols in C Language. 0 to 47, 58 to 64, 91 to 96, 123 to 127.

134. Expand or Abbreviate ASCII with regard to C Language.

a. Australian Standard Code for Information Interchange
b. American Standard Code for Information Interchange
c. American Symbolic Code for Information Interchange
d. Australian Symbolic Code for Information Interchange

Explanation:

There were only 128 Characters with 7 Bits in Original ASCII specification. Present character standard in all modern programming languages is UNICODE which covers all languages, Emojis and other special symbols all over the world.

135. What is the output of C Program with Switch Statement.?

```
int main()
{
   int a=5;
   switch(a)
   {
      case 0: printf("0 ");
      case 3: printf("3 ");
      case 5: printf("5 ");
      default: printf("RABBIT ");
```

```
    }
    a=10;
    switch(a)
    {
       case 0: printf("0 ");
       case 3: printf("3 ");
       case 5: printf("5 ");
       default: printf("RABBIT "); break;
    }
    return 0;
}
```

a. 5 RABBIT

b. 0 3 5 RABBIT 0 3 5 RABBIT

c. 0 3 5 RABBIT RABBIT

d. 3 5 RABBIT RABBIT

Explanation:

Absence of break; after case statement causes control to go to next case automatically. So after matching 3 with a==3, program prints 3 and control falls down the ladder without checking case again printing everything below it. So Switch checks only once and falls down.

136. What is the output of C Program with switch statement.?

```
int main()
{
   int a=3;
   switch(a)
   {
      case 2: printf("ZERO "); break;

      case default: printf("RABBIT ");
   }
  }
```

a. RABBIT

b. ZERO RABBIT

c. No output

d. Compiler error

Explanation:

Notice that it is "default" not "case default".

137. What is the output of C Program with switch statement or block.?

```
int main()
{
  int a=3;
  switch(a)
  {

  }
  printf("MySwitch");
}
```

a. MySwitch

b. No Output

c. Compiler Error

d. None of the above

Explanation:

Yes. It is okay to omit CASE: statements inside SWITCH construct or block.

138. What is the output of C Program with switch statement or block.?

```
int main()
{
  int a;

  switch(a)
  {
```

```
        printf("APACHE ");
    }
    printf("HEROHONDA");
}
```

a. APACHE HEROHONDA

b. HEROHONDA

c. No Output

d. Compiler error

Explanation:

Notice the missing CASE or DEFAULT statements. Still compiler accepts. But without CASE statement nothing will be printed inside of SWITCH.

139. What is the output of C program with switch statement or block.?

```
int main()
{
    int a;
    switch(a);
    {
        printf("DEER ");
    }
    printf("LION");
}
```

a. LION

b. DEER LION

c. Compiler error

d. None of the above

Explanation:

Notice a semicolon at the end of switch(a);. So, printf DEER is out of SWITCH.

140. What is the output of C program with switch statement or block.?

```
int main()
{
   char code='K';
   switch(code)
   {
      case 'A': printf("ANT ");break;
      case 'K': printf("KING "); break;
      default: printf("NOKING");
   }
   printf("PALACE");
}
```

a. KING PALACE

b. KING NOTHING PALACE

c. ANT KING PALACE

d. Compiler error for using Non Integers as CASE constants.

Explanation:

Any character is replaced by ASCII code or number by the compiler. So it is allowed to use CHARACTER constants for CASE constants.

141. What is the output of C Program with switch statement or block.?

```
int main()
{
   char code='K';
   switch(code)
   {
      case "A": printf("ANT ");break;
      case "K": printf("KING "); break;
  default: printf("NOKING");
   }

   printf("PALACE");
}
```

a. ANT KING PALACE

b. KING PALACE

c. PALACE

d. Compiler error

Explanation:

You can not use STRING constants with DOUBLE QUOTES as CASE constants. Only expressions leading or constants leading to Integers are allowed.

142. Choose a correct statement about a C Switch Construct.

a. default case is optional inside switch.

b. break; causes the control to exit the switch immediately and avoid fall down to other CASE statements.

c. You can not use duplicate CASE Constants inside a Switch construct.

d. All the above.

143. Choose a correct statement about C language arrays.

a. An array address is the address of first element of array itself.

b. An array size must be declared if not initialized immediately.

c. Array size is the sum of sizes of all elements of the array.

d. All the above

144. What are the Types of Arrays.?

a. int, long, float, double

b. struct, enum

c. char

d. All the above

145. An array Index starts with.?

a. -1

b. 0

c. 1

d. 2

146. Choose a correct statement about C language arrays.

a. An array size can not changed once it is created.

b. Array element value can be changed any number of times

c. To access Nth element of an array students, use students[n-1] as the starting index is 0.

d. All the above

147. What is the output of C Program.? int main() { int a[]; a[4] = {1,2,3,4}; printf("%d", a[0]); }

a. 1

b. 2

c. 4

d. Compiler error

Explanation:

If you do not initialize an array, you must mention ARRAY SIZE.

148. What is the output of C Program.? int main() { int a[] = {1,2,3,4}; int b[4] = {5,6,7,8}; printf("%d,%d", a[0], b[0]); }

a. 1,5

b. 2,6

c. 0 0

d. Compiler error

Explanation:

It is perfectly allowed to skip array size if you are initializing at the same time. a[0] is first element.

int a[] = {1,2,3,4};

149. What is an array Base Address in C language.?
   a. Base address is the address of 0th index element.
   b. An array b[] base address is &b[0]
   c. An array b[] base address can be printed with printf("%d", b);
   d. All the above

150. In computer science, algorithm refers to a special method usable by a computer for the solution to a problem.
   a. True
   b. False

   Explanation: The statement is true. This word algorithm refers to a special method usable by a computer for the solution to a problem. The statement of the problem specifies in general terms the desired input/output relationship.

151. The symbol denotes _____

   a. I/O
   b. Flow
   c. Terminal
   d. Decision

   Explanation: The symbol denotes a terminal. It is used for indication of start and stop nodes of a program.

152. In computer science, algorithm refers to a pictorial representation of a flowchart.
   a. True
   b. False

   Explanation: The statement is false. The correct statement would be: In computer science, flowchart refers to a pictorial representation of an algorithm.

153. The process of drawing a flowchart for an algorithm is called _____
   a. Performance
   b. Evaluation

c. Algorithmic Representation

d. Flowcharting

Explanation: It is called as flowcharting. A flowchart is nothing but a pictorial representation of an algorithm.

154. Actual instructions in flowcharting are represented in _____

a. Circles

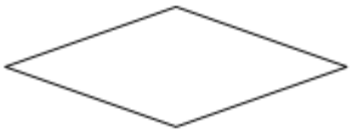
b. Boxes

c. Arrows

d. Lines

Explanation: The actual instructions are written in boxes. Boxes are connected by using arrows to indicate the exact flow of a flowchart and the order in which they are to be executed.

155. The following box denotes?



a. Decision

b. Initiation

c. Initialization

d. I/O

Explanation: A diamond shape box denotes the decision making statements. It jumps to a truth value or a false value.

156. A box that can represent two different conditions.

a. Rectangle

b. Diamond

c. Circle

d. Parallelogram

Explanation: A diamond shape box denotes either a truth value or a false value. It jumps onto two different statements following it via flow lines.

157. There should be certain set standards on the amount of details that should be provided in a flowchart.
   a. True
   b. False

   Explanation: The statement is false. There should be no set standards on the amount of details that should be provided in a flowchart.

158. A detailed flowchart is called _____
   a. Stack
   b. Macro
   c. Micro
   d. Union

   Explanation: A detailed flowchart or a flowchart with more details is called as micro flowchart. It represents all the components of the algorithm that is followed.

159. Which of the following is not an advantage of a flowchart?
   a. Better communication
   b. Efficient coding
   c. Systematic testing
   d. Improper documentation

   Explanation: Flowcharts provide a proper documentation. It also provides systematic debugging.

160. A flowchart that outlines the main segments of a program.
   a. Queue
   b. Macro
   c. Micro
   d. Union

Explanation: The answer is Macro Flowchart. A macro flowchart outlines the important components of a program. It therefore shows fewer details.

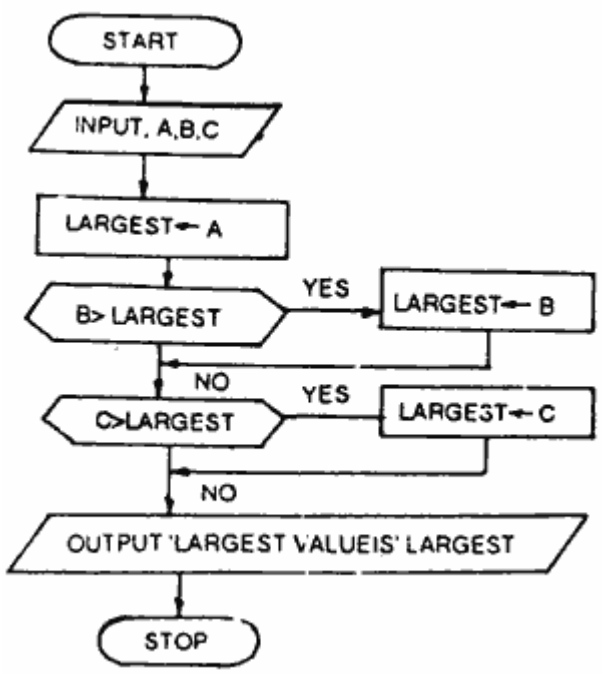
161. The symbol shown in the Figure in flow chart represents.

a. In-connector
b. Out-connector
c. Output
d. End

162. The symbol shown in Figure in flow chart represents

a. In connector
b. Out connector
c. Annotation
d. Input/output

Question 163 and 164 refer to flow chart shown in Figure.



163. The problem represented by the flow chart is
    a. To compare A, B and C
    b. To find lowest of A, B and C
    c. To find average of A,B and C
    d. To find largest value of A, B and C

164. The output will be
    a. Stop
    b. Average number
    c. Largest number
    d. Least number.

165. This characteristic often draws the line between what is feasible and what is impossible.
    a. Performance
    b. System Evaluation
    c. Modularity
    d. Reliability

    Explanation: Algorithms help us to understand scalability. Performance often draws the line between what is feasible and what is impossible.

166. Which of the following is incorrect?Algorithms can be represented:
    a. as pseudo codes
    b. as syntax
    c. as programs
    d. as flowcharts

Explanation: Representation of algorithms:
-As programs
-As flowcharts
-As pseudo codes.

167. When an algorithm is written in the form of a programming language, it becomes a _____

a. Flowchart

b. Program

c. Pseudo code

d. Syntax

Explanation: An algorithm becomes a program when it is written in the form of a programming language. Thus, any program is an algorithm.

168. Any algorithm is a program.

a. True

b. False

Explanation: The statement is false. An algorithm is represented in the form of a programming language is called a program. Any program is an algorithm but the reverse is not true.

169. Which of the following is not a valid variable name declaration?

a. int __a3;

b. int __3a;

c. int __A3;

d. None of the mentioned

170. Which of the following is not a valid variable name declaration?

a. int _a3;

b. int a_3;

c. int 3_a;

d. int _3a

Explanation: Variable name cannot start with a digit.

171. Why do variable names beginning with the underscore is not encouraged?

a. It is not standardized.

b. To avoid conflicts since assemblers and loaders use such names

c. To avoid conflicts since library routines use such names

d. To avoid conflicts with environment variables of an operating system

172. All keywords in C are in _____
   a. LowerCase letters
   b. UpperCase letters
   c. CamelCase letters
   d. None of the mentioned

173. Variable name resolution (number of significant characters for the uniqueness of variable) depends on _____
   a. Compiler and linker implementations
   b. Assemblers and loaders implementations
   c. C language
   d. None of the mentioned

   Explanation: It depends on the standard to which compiler and linkers are adhering.

174. Which of the following is not a valid C variable name?
   a. int number;
   b. float rate;
   c. int variable_count;
   d. int $main;

   Explanation: Since only underscore and no other special character is allowed in a variable name, it results in an error.

175. Which of the following is true for variable names in C?
   a. They can contain alphanumeric characters as well as special characters
   b. It is not an error to declare a variable to be one of the keywords(like goto, static)
   c. Variable names cannot start with a digit
   d. Variable can be of any length

   Explanation: According to the syntax for C variable name, it cannot start with a digit.

176. Which is valid C expression?
   a. int my_num = 100,000;
   b. int my_num = 100000;

c. int my num = 1000;

d. int $my_num = 10000;

Explanation: Space, comma and $ cannot be used in a variable name.

177. What will be the output of the following C code?

```
#include <stdio.h>
  int main()
    {
       printf("Hello World! %d \n", x);
       return 0;
     }
```

a. Hello World! x;

b. Hello World! followed by a junk value

c. Compile time error

d. Hello World!

Explanation: It results in an error since x is used without declaring the variable x.

Output:

$ cc pgm1.c

pgm1.c: In function ‘main’:

pgm1.c:4: error: ‘x’ undeclared (first use in this function)

pgm1.c:4: error: (Each undeclared identifier is reported only once

pgm1.c:4: error: for each function it appears in.)

178. What will be the output of the following C code?

```
#include <stdio.h>
  int main()
  {
     int y = 10000;
     int y = 34;
     printf("Hello World! %d\n", y);
    return 0;
  }
```

a. Compile time error

b. Hello World! 34

c. Hello World! 1000

d. Hello World! followed by a junk value

Explanation: Since y is already defined, redefining it results in an error.

Output:

$ cc pgm2.c

pgm2.c: In function ‘main’:

pgm2.c:5: error: redefinition of ‘y’

pgm2.c:4: note: previous definition of ‘y’ was here

179. Which of the following is not a valid variable name declaration?

a. float PI = 3.14;

b. double PI = 3.14;

c. int PI = 3.14;

d. #define PI 3.14

Explanation: #define PI 3.14 is a macro preprocessor, it is a textual substitution.

180. What will happen if the following C code is executed?

```
#include <stdio.h>
 int main()
 {
    int main = 3;
    printf("%d", main);
    return 0;
 }
```

a. It will cause a compile-time error

b. It will cause a run-time error

c. It will run without any error and prints 3

d. It will experience infinite looping

Explanation: A C program can have same function name and same variable name.

$ cc pgm3.c

$ a.out

3

181. What is the problem in the following variable declaration?

float 3Bedroom-Hall-Kitchen?;

a. The variable name begins with an integer
b. The special character '-'
c. The special character '?'
d. All of the mentioned

Explanation: A variable name cannot start with an integer, along with that the C compiler interprets the '-' and '?' as a minus operator and a question mark operator respectively.

182. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
  int ThisIsVariableName = 12;
   int ThisIsVariablename = 14;
   printf("%d", ThisIsVariablename);
  return 0;
}
```

a. The program will print 12
b. The program will print 14
c. The program will have a runtime error
d. The program will cause a compile-time error due to redeclaration

Explanation: Variable names ThisIsVariablename and ThisIsVariableName are both distinct as C is case sensitive.

Output:

```
$ cc pgm4.c
$ a.out
14
```

183. Which of the following cannot be a variable name in C?
   a. volatile
   b. true
   c. friend
   d. export

Explanation: volatile is C keyword.

184. The format identifier '%i' is also used for _____ data type.
   a. char
   b. int
   c. float
   d. double

Explanation: Both %d and %i can be used as a format identifier for int data type.

185. Which data type is most suitable for storing a number 65000 in a 32-bit system?
   a. signed short
   b. unsigned short
   c. long
   d. int

Explanation: 65000 comes in the range of short (16-bit) which occupies the least memory. Signed short ranges from -32768 to 32767 and hence we should use unsigned short.

186. What is the size of an int data type?
   a. 4 Bytes
   b. 8 Bytes
   c. Depends on the system/compiler

d. Cannot be determined

Explanation: The size of the data types depend on the system.

187. Which is correct with respect to the size of the data types?

a. char > int > float

b. int > char > float

c. char < int < double

d. double > char > int

Explanation: char has less bytes than int and int has less bytes than double in any system

188. What will be the output of the following C code?

```
#include <stdio.h>
    int main()
{
        float x = 'a';
        printf("%f", x);
        return 0;
    }
```

a. a

b. run time error

c. a.0000000

d. 97.000000

Explanation: Since the ASCII value of a is 97, the same is assigned to the float variable and printed.

189. Which of the data types has the size that is variable?

a. int

b. struct

c. float

d. double

Explanation: Since the size of the structure depends on its fields, it has a variable size.

190. What will be the output of the following C code?

```
#include <stdio.h>
  int main()
  {
     j = 10;
     printf("%d\n", j++);
   return 0;
  }
```

a. 10

b. 11

c. Compile time error

d. 0

Explanation: Variable j is not defined.

191. Will the following C code compile without any error?

```
#include <stdio.h>
 int main()
 {
    for (int k = 0; k < 10; k++);
       return 0;
 }
```

a. Yes

b. No

c. Depends on the C standard implemented by compilers

d. Error

Explanation: Compilers implementing C90 do not allow this, but compilers implementing C99 allow it.

192. Will the following C code compile without any error?

```
#include <stdio.h>
  int main()
  {
     int k;
     {
```

```
        int k;
        for (k = 0; k < 10; k++);
    }
}
```

a. Yes

b. No

c. Depends on the compiler

d. Depends on the C standard implemented by compilers

Explanation: There can be blocks inside the block. But within a block, variables have only block scope.

193. What will be the output of the following C code?

```
#include <stdio.h>
  int main()
  {
     int i = -3;
     int k = i % 2;
     printf("%d\n", k);
  }
```

a. Compile time error

b. -1

c. 1

d. Implementation defined

194. What will be the output of the following C code?

```
#include <stdio.h>
  int main()
  {
     int i = 3;
     int l = i / -2;
     int k = i % -2;
     printf("%d %d\n", l, k);
     return 0;
  }
```

a. Compile time error

b. -1 1

c. 1 -1

d. Implementation defined

195. What will be the output of the following C code?

```
#include <stdio.h>
  int main()
  {
     int i = 5;
     i = i / 3;
     printf("%d\n", i);
    return 0;
  }
```

a. Implementation defined

b. 1

c. 3

d. Compile time error

196. What will be the final value of x in the following C code?

```
#include <stdio.h>
   void main()
   {
      int x = 5 * 9 / 3 + 9;
   }
```

a. 3.75

b. Depends on compiler

c. 24

d. 3

197. What will be the output of the following C code?

```
#include <stdio.h>
  void main()
  {
```

```
    int x = 5.3 % 2;
    printf("Value of x is %d", x);
  }
```

a. Value of x is 2.3

b. Value of x is 1

c. Value of x is 0.3

d. Compile time error

198. What will be the output of the following C code?

```
#include <stdio.h>
  void main()
  {
    int y = 3;
    int x = 5 % 2 * 3 / 2;
    printf("Value of x is %d", x);
  }
```

a. Value of x is 1

b. Value of x is 2

c. Value of x is 3

d. Compile time error

199. What will be the output of the following C code?

```
#include <stdio.h>
  void main()
  {
    int a = 3;
    int b = ++a + a++ + --a;
    printf("Value of b is %d", b);
  }
```

a. Value of x is 12

b. Value of x is 13

c. Value of x is 10

d. Undefined behaviour

200. What is the precedence of arithmetic operators (from highest to lowest)?
   a. %, *, /, +, –
   b. %, +, /, *, –
   c. +, -, %, *, /
   d. %, +, -, *, /

201. Which of the following is not an arithmetic operation?
   a. a * = 10;
   b. a / = 10;
   c. a ! = 10;
   d. a % = 10;

202. Which of the following data type will throw an error on modulus operation(%)?
   a. char
   b. short
   c. int
   d. float

203. Which among the following are the fundamental arithmetic operators, i.e, performing the desired operation can be done using that operator only?
   a. +, –
   b. +, -, %
   c. +, -, *, /
   d. +, -, *, /, %

204. What will be the output of the following C code?

```
#include <stdio.h>
 int main()
 {
    int a = 10;
    double b = 5.6;
    int c;
    c = a + b;
```

```
    printf("%d", c);
}
```

a. 15

b. 16

c. 15.6

d. 10