| | | | | |
|---|---|---|---|---|
| 1. | **Pentium computer is an example for** | | | |
| | A) | Notebook | B) | Microcomputer |
| | C) | Hybrid Computer | D) | Supercomputer |
| 2. | **A Computer program that converts assembly language to machine language is called** | | | |
| | A) | Compiler | B) | Linker |
| | C) | Assembler | D) | Translator |
| 3. | **FORTRAN is an example of** | | | |
| | A) | High level language | B) | Machine language |
| | C) | Assembly language | D) | Compiler language |
| 4. | **Information provided by user to computer is called** | | | |
| | A) | Data | B) | Output |
| | C) | Compute | D) | |
| 5. | **A computer in which data is recognized as a continuous measurement of a physical property like voltage is called** | | | |
| | A) | Supercomputer | B) | Laptop |
| | C) | Analog computer | D) | Hybrid computer |
| 6. | **Computers used for a specific job are called** | | | |
| | A) | Hybrid computers | B) | Digital computers |
| | C) | Special purpose computers | D) | Mainframe |
| 7. | **PARAM is an example of** | | | |
| | A) | Super computer | B) | Mainframe |
| | C) | Hybrid computer | D) | Notebook |
| 8. | **Step by step procedure to solve a give problem is called** | | | |
| | A) | Flowchart | B) | Algorithm |
| | C) | Machine program | D) | Compiler |
| 9. | **The ways that specific words and symbols are used by each language is called** | | | |
| | A) | Program | B) | Algorithm |
| | C) | Syntax | D) | Grammar |
| 10. | **The process of putting together all the external references that are required by the program is called** | | | |
| | A) | Translation | B) | Compilation |
| | C) | Coding | D) | Linking |
| 11. | **The process of correcting errors in a program is called** | | | |
| | A) | Debugging | B) | Testing |
| | C) | Coding | D) | Linking |
| 12. | **Why does a Machine language program run faster?** | | | |
| | A) | It's easy to program | B) | No Translation is needed |
| | C) | Easy to debug | D) | Debugging is easy |
| 13. | **Which of the following is an example for IDE** | | | |
| | A) | Assembler | A) | Compiler |
| | C) | Code Blocks | C) | Linux |
| 14. | **The interface between hardware and user is** | | | |
| | A) | Editor | B) | OS |
| | C) | File | D) | Pointer |
| 15. | **Which of the following is an Input device?** | | | |

| | | | | |
|---|---|---|---|---|
| | A) | Keyboard | B) | Speaker |
| | C) | Printer | D) | CPU |

| | | | | |
|---|---|---|---|---|
| 1. | **Which of the following is not a valid C variable name?** | | | |
| | A) | int number; | B) | float rate; |
| | C) | int variable_count; | D) | int $main; |
| 2. | **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 |
| | C) | Variable names cannot start with a digit | D) | Variable can be of any |
| 3. | **Which keyword is used to prevent any changes in the variable within a C program?** | | | |
| | A) | immutable | B) | mutable |
| | C) | const | D) | volatile |
| 4. | **Property which allows to produce different executable for different platforms in C is called?** | | | |
| | A) | File inclusion | B) | Selective inclusion |
| | C) | Conditional compilation | D) | Recursive macros |
| 5. | **What is #include <stdio.h>?** | | | |
| | A) | Preprocessor directive | B) | Inclusion directive |
| | C) | File inclusion directive | D) | None of the |
| 6. | **The C-preprocessors are specified with____symbol.** | | | |
| | A) | # | B) | $ |
| | C) | “ “ | D) | & |
| 7. | **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 |

| | | | | |
|---|---|---|---|---|
| **8.** | **Which of the following is not a data type?** | | | |
| | A) | Symbolic Data | B) | Alphanumeric Data |
| | C) | Numeric Data | D) | Alphabetic Data |
| **9.** | **What are the entities whose values can be changed called?** | | | |
| | A) | Constants | B) | Variables |
| | C) | Modules | D) | Tokens |
| **10.** | **Which of the following are tokens in C?** | | | |
| | A) | Keywords | B) | Variables |
| | C) | Constants | D) | All of the above |
| **11.** | **C programs are converted into machine language with the help of** | | | |
| | A) | An Editor | B) | A compiler |
| | C) | An operating system | D) | None of the above |
| **12.** | **What is C Tokens?** | | | |
| | A) | The smallest individual units of c program | B) | The basic element recognized by the compiler |
| | C) | The largest individual units of program | D) | A & B Both |
| **13.** | **What is Keywords?** | | | |
| | A) | Keywords have some predefine meanings and these meanings can be changed. | A) | Keywords have some unknown meanings and these meanings cannot be changed. |
| | C) | Keywords have some predefine meanings and these meanings cannot be changed. | C) | None of the above |

| | | | | |
|---|---|---|---|---|
| **14.** | **What is constant?** | | | |
| | A) | Constants have fixed values that do not change during the execution of a program | B) | Constants have fixed values that change during the execution of a program |
| | C) | Constants have unknown values that may be change during the execution of a program | D) | None of the above |
| **15.** | **Which escape character can be used to begin a new line in C?** | | | |
| | A) | \a | B) | \b |
| | C) | \m | D) | \n |

| | | | | |
|---|---|---|---|---|
| **1.** | **Which arithmetic operator in C is used to find the remainder of division operation** | | | |
| | A) | **%** | B) | / |
| | C) | mod | D) | rem |
| **2.** | **What is the value of 'y' after the evaluating the following expression?**<br>**y=5+(-4)*6/3%2** | | | |
| | A) | 1 | B) | 2 |
| | C) | 0 | D) | **5** |
| **3.** | **What is the value of 'z' after the following code is executed?**<br>**int x = 3, z;**<br>**z = x++ + ++x;** | | | |
| | A) | 7 | B) | **8** |
| | C) | 9 | D) | 10 |
| **4.** | **What is the value of 'result' after the following code is executed?**<br>**int num = 20, result;**<br>**result = (num % 2 == 0) ? num / 2 : num * 2;** | | | |
| | A) | **10** | B) | 20 |
| | C) | 40 | D) | 60 |
| **5.** | **What is the value of 'y' after the following code is executed?**<br>**int x = 3, y;**<br>**y=x/++x;** | | | |
| | A) | 2 | B) | 0 |
| | C) | **1** | D) | 3 |
| **6.** | **What is the value of 'y' after the following code is executed?**<br>**int x = 10, y=20;**<br>**x=!x;**<br>**y=!!x \|\| (y =!!y);** | | | |
| | A) | 0 | B) | **1** |
| | C) | 10 | D) | 20 |
| **7.** | **When an expression involves following types of operators, what is the order of evaluation?**<br>**1. Relational**<br>**2. Logical**<br>**3. Assignment** | | | |
| | A) | **1,2,3** | B) | 2,1,3 |
| | C) | 3,2,1 | D) | 1,3,2 |
| **8.** | **What is the value of 'a' after the following code is executed?**<br>**int a = 5, y;**<br>**y=a-- + ++a;** | | | |

| | | | | |
|---|---|---|---|---|
| | A) | **5** | B) | 6 |
| | C) | 4 | D) | 11 |
| **9.** | **What is the value of 'a' after the following code is executed?**<br>**int x=8, a=4, b=3, c=2;**<br>**a=--a*x-b/a++*c;** | | | |
| | A) | 20 | B) | 24 |
| | C) | **23** | D) | 22 |
| **10.** | **Which expression extracts last digit from a three-digit number?** | | | |
| | A) | **num%10** | B) | num/10 |
| | C) | num/100 | D) | num%100 |
| **11.** | **What is the equivalent arithmetic expression for bitwise expression n<<1?** | | | |
| | A) | n+2 | B) | n-2 |
| | C) | **n*2** | D) | n/2 |
| **12.** | **What is the value of 'z' after the following code is executed?**<br>**int x = 3, y=3, z=3;**<br>**z-=-x-- - --y;** | | | |
| | A) | 5 | B) | 6 |
| | C) | 7 | D) | **8** |
| **13.** | **What is the value of 'j' after the following code is executed?**<br>**int i = 3, j;**<br>**j=++i*++i*++i;** | | | |
| | A) | 120 | A) | **216** |
| | C) | 126 | C) | Error |
| **14.** | **What is the value of 'z' after the following code is executed?**<br>**int x = 3, y=4, z=4;**<br>**z=(z>=y>=x)?100:200** | | | |
| | A) | 100 | B) | **200** |
| | C) | 1 | D) | 0 |
| **15.** | **What is the value of 'z' after the following code is executed?**<br>**int x = -1, y=1, z=2;**<br>**z=++x && ++y \|\| ++z;** | | | |
| | A) | **1** | B) | 0 |
| | C) | 3 | D) | Error |
| **16.** | **Which of the following is an example of a compounded assignment statement?** | | | |
| | A) | a=5 | B) | a+=5 |
| | C) | a=b=c | D) | a=b |
| **17.** | **The associativity of ! Operator is** | | | |
| | A) | **Right to Left** | B) | Left to Right |
| | C) | Either A or B | D) | None |
| **18.** | **Which operator has the lowest priority?** | | | |
| | A) | ++ | B) | % |
| | C) | + | D) | **\|\|** |
| **19.** | **Which operator has the highest priority?** | | | |
| | A) | **++** | B) | % |
| | C) | + | D) | \|\| |
| **20.** | **Operators have precedence. Precedence determines which operator is** | | | |
| | A) | faster | B) | takes less memory |
| | C) | **evaluated first** | D) | takes no arguments |
| **21.** | **Integer Division results in** | | | |
| | A) | Rounding the fractional part | B) | **Truncating the fractional part** |
| | C) | Floating value | D) | Reminder |
| **22.** | **What will be the output of the expression 11 ^ 5 ^ 2?** | | | |
| | A) | **12** | B) | 3 |

| | | | | |
|---|---|---|---|---|
| | C) | 6 | D) | 11 |
| **23.** | **The type cast operator is** | | | |
| | A) | sizeof() | B) | **(type)** |
| | C) | cast() | D) | None |
| **24.** | **The operator + in a+=4 means** | | | |
| | A) | **a=a+4** | B) | a+4=a |
| | C) | a=4 | D) | a=4+4 |

| | | | | |
|---|---|---|---|---|
| **1.** | **Which of the following operators is used to perform a bitwise right shift?** | | | |
| | A) | << | B) | >> |
| | C) | & | D) | \| |
| **2.** | **If x = 5, what will be the value of x << 2?** | | | |
| | A) | 10 | B) | 20 |
| | C) | 15 | D) | 25 |
| **3.** | **What will be the output of the following C code?**<br>#include <stdio.h> void main()<br>{<br>int k = 8; int m = 7;<br>k < m ? k++ : m = k;<br>printf("%d", k);<br>} | | | |
| | A) | 7 | B) | 8 |
| | C) | Compile time error | D) | Run time error |
| **4.** | **What will be the data type of the following expression? (Initial data type: a = int, var1= double, var2 = float)**<br>expression (a < 50)? var1 : var2; | | | |
| | A) | int | B) | float |
| | C) | double | D) | Cannot be determined |
| **5.** | **Which of the following arithmetic operator takes only integer operands?** | | | |
| | A) | + | B) | - |
| | C) | / | D) | % |
| **6.** | **Which of the following operator has the lowest priority?** | | | |
| | A) | && | B) | + |
| | C) | * | D) | !== |
| **7.** | **What will be the output of the following C code?**<br>#include <stdio.h><br>int main() | | | |

| | | | | |
|---|---|---|---|---|
| | { int n = 8; n = n / 3; printf("%d", n); return 0; } | | | |
| | A) | **2** | B) | 1 |
| | C) | 3 | D) | Compile-time error |
| **8.** | **What will be the result of the following expression if a = 6, b = 3, and c = 4?** (a * b - c) / (b + c % a) + (a % b * c) | | | |
| | A) | 7 | B) | 8 |
| | C) | 6 | D) | 10 |
| **9.** | **What will be the result of the following expression if x = 10 and y = 4?** (x / y) == 2 && (x % y) > 0 | | | |
| | A) | True | B) | False |
| | C) | 10 | D) | 20 |
| **10.** | **What will be the result of the following expression if p = 7 and q = 10?** (p + q) > 15 \|\| (p - q) < 0 | | | |
| | A) | True | B) | False |
| | C) | 15 | D) | 10 |
| **11.** | **What will be the output of the following code snippet using the sizeof operator?** #include <stdio.h> int main() { int a; double b; printf("%lu %lu", sizeof(a), sizeof(b)); return 0; } | | | |
| | A) | 4 8 | B) | 8 4 |
| | C) | 4 4 | D) | 8 8 |
| **12.** | **What will be the result of the following C code snippet using the comma operator?** #include <stdio.h> int main() { int x = 1, y = 2; int z = (x++, y++); printf("%d %d %d", x, y, z); return 0; } | | | |
| | A) | 2 3 2 | B) | 2 3 3 |
| | C) | 1 2 1 | D) | 2 2 2 |
| **13.** | **What will be the result of the following C expression?** | | | |

| | | | | |
|---|---|---|---|---|
| | int x = 15; int y = 9;<br>int z = x ^ y;//bitwise xor | | | |
| | A) | 6 | A) | 12 |
| | C) | 5 | C) | 3 |
| **14.** | **Which of the following operators is right-associative?**<br>a) + (addition)<br>a) * (multiplication)<br>b) - (subtraction)<br>c) ?: (ternary conditional) | | | |
| | A) | + | B) | * |
| | C) | - | D) | ?: |

| | | | | |
|---|---|---|---|---|
| **1.** | What is the purpose of an if statement in C? | | | |
| | A) | To iterate over a block of code | B) | To declare a variable |
| | C) | To execute a block of code conditionally | D) | To define a function |
| **2.** | What is a switch case statement used for in C? | | | |
| | A) | To define constants | B) | To perform different actions based on different conditions |
| | C) | To create loops | D) | To declare variables |
| **3.** | How can multiple conditions be combined in an if statement in C? | | | |
| | A) | Using the '+' operator | B) | Using the '&&' and '\|\|' operators |
| | C) | By nesting if statements | D) | By separating conditions with commas |
| **4.** | Can the else if ladder be used instead of the switch case in C? | | | |
| | A) | Yes | B) | No |
| | C) | Only in certain cases | D) | Only if there are less than 3 cases |
| **5.** | #include <stdio.h> | | | |

```
void main()
{
   char *ch;
   printf("enter a value between 1 to 3”);
   scanf("%s", ch); switch (ch)
   {
case "1": printf("1"); break;
case "2": printf("2"); break;
}
}
```

| | | | |
|---|---|---|---|
| A) | 1 | B) | 2 |
| C) | **Compile time error** | D) | No compile time error |

**6.** is the built in multiway decision statement in C.

| | | | |
|---|---|---|---|
| A) | for | B) | switch |
| C) | if | D) | while |

**7.** What will be the output of the following program?

```
#include<stdio.h> int main()
{
   int a=300, b, c; if(a>=400)
     b=300; c=200;
   printf(“%d%d\n”, b, c); return 0;
}
```

| | | | |
|---|---|---|---|
| A) | Garbage value, Garbage value | B) | 300, 200 |
| C) | 200, 300 | D) | Garbage value, 200 |

**8.** What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input)

```
#include <stdio.h>
 void main()
{
   int ch;
   printf("enter a value between 1 to 2:"); scanf("%d", &ch);
   switch (ch, ch + 1)
```

```
{
  case 1: printf("1\n"); break;
  case 2: printf("2");
}
}
```

| | | | |
|---|---|---|---|
| A) | 1 | B) | 2 |
| C) | 3 | D) | Run time error |

**9.** Which one is true about the break statement in a switch case statement?

| | | | |
|---|---|---|---|
| A) | To skip the current case | B) | To end the program |
| C) | To exit the switch case statement | D) | To exit the loop |

**10.** Read following program and find output of the following code? int p = 15;

```
if (p > 3)
{
printf("p is greater than 3\n");
}
else
{
printf("p is not greater than 3\n");
}
```

A) B) C) D)

| | | | |
|---|---|---|---|
| A) | p is greater than 3 | B) | p is not greater than 3 |
| C) | 15 | D) | Compilation error |

**11.** In a switch statement, what happens if no break statement is used after a case block?

| | | | |
|---|---|---|---|
| A) | The program will terminate | B) | The next case block will be executed |
| C) | The switch statement will end | D) | The default case will be executed |

**12.** Which of the following statements is true regarding nested if statements?

| | | | |
|---|---|---|---|
| A) | Nested if statements can have multiple else blocks. | B) | Nested if statements are not allowed in C. |
| C) | The else block must match the nearest preceding if statement. | D) | Nested if statements must always have a default block. |

| | | | | |
|---|---|---|---|---|
| **13.** | What is the correct syntax for an else if ladder in C? | | | |
| | A) | else (condition) { statements } | A) | if (condition) { statements } else if { statements } |
| | C) | if (condition) { statements } else if (condition) { statements }else{ statements } | C) | else if (condition) statements; |
| **14.** | 7) What will be the output of the following C code?<br>int x = 3;<br>switch (x) { case 1:<br>printf("One"); break;<br>case 2:<br>printf("Two"); break;<br>case 3:<br>printf("Three");<br>default:<br>printf("Default case");<br>} | | | |
| | A) | One | B) | Two |
| | C) | Three | D) | Default case |
| **15.** | What happens when none of the case values match the expression in a switch statement, and no default case is provided? | | | |
| | A) | The first case is executed by default. | B) | The switch statement is ignored, and the next statement after switch is executed. |
| | C) | A compilation error occurs. | D) | The program terminates. |

| | | | | |
|---|---|---|---|---|
| **1.** | Which of the following statements is true about the goto statement in C? | | | |
| | A) | It can be used to jump to a specific line in a function. | B) | It can be used to jump to a label within a different function. |
| | C) | It can be used to jump to a specific location in the code by using a label. | D) | It can be used to exit from a loop or switch statement. |

**2.**

```
#include <stdio.h> int main()
{
      int i = 0; while (i < 5)
     {
        if (i == 2)
        {
            break;
        }
       printf("%d ", i); i++;
     }
    return 0;
}
```

What is the output of the code?

| | | | |
|---|---|---|---|
| A) | 0 1 2 | B) | 0 1 |
| C) | 0 1 2 3 4 | D) | 0 1 2 3 |

**3.** In which scenario would you use the continue statement?

| | | | |
|---|---|---|---|
| A) | To exit from a loop early and execute the code after the loop. | B) | To skip the remaining statements in the current iteration of a loop and proceed to the next iteration. |
| C) | To jump to a specific line of code unconditionally. | D) | To break out of a loop or switch statement. |

**4.**

```
int main()
{
   int x = 0; while (x < 5)
   {
      if (x == 3)
      {
         x++;
         continue;
      }
      printf("%d ", x); x++;
   }
   return 0;
}
```

What is the output of the code?

| | | | |
|---|---|---|---|
| A) | 0 1 2 3 4 | B) | 0 1 2 4 |
| C) | 0 1 2 3 | D) | 0 1 2 3 4 5 |

**5.**

```
#include <stdio.h>

int main() { int i;
   for (i = 0; i < 10; i++) { if (i == 5) {
         break;
      }
      printf("%d ", i);
   }
```

```
    printf("Loop ended.\n"); return 0;
}
```

What is the output of the code?

| | | | |
|---|---|---|---|
| A) | 0 1 2 3 4 5 Loop ended | B) | 0 1 2 3 4 Loop ended |
| C) | 0 1 2 3 4 5 6 7 8 9 Loop ended | D) | Loop ended |

**6.**

```
#include <stdio.h>


    int main() { int
    if (i == 3) { break;
    }
    printf("%d ", i);
  }
  return 0;
}
```

What is the output of the code?

| | | | |
|---|---|---|---|
| A) | 0 1 2 3 | B) | 0 1 2 |
| C) | 1 2 3 | D) | 0 1 2 3 4 |

**7.** Which of the following loops is guaranteed to execute at least once?

| | | | |
|---|---|---|---|
| A) | for loop | B) | while loop |
| C) | do-while loop | D) | None |

**8.**

```
#include <stdio.h>

int main() { int i;
  for (i = 0; i < 3; i++) { printf("%d ", i);
    if (i == 1) { continue;
    }
    printf("Hello ");
  }
  return 0;
}
```

What is the output of the code?

| | | | |
|---|---|---|---|
| A) | 0 Hello 1 2 Hello | B) | 0 Hello 1 2 |
| C) | 0 Hello 1 Hello 2 Hello | D) | 0 Hello 1 Hello 2 Hello 3 Hello |

**9.** How many times will the following do-while loop execute?

```
#include <stdio.h>

int main() { int i = 5; do {
    i++;
  } while (i < 5); printf("%d\n", i); return 0;
}
```

| | | | | |
|---|---|---|---|---|
| | What is the output of the code? | | | |
| | A) | 0 | B) | 1 |
| | C) | 5 | D) | Infinite times |
| **10.** | #include <stdio.h><br><br>int main() { int i;<br>for (i = 3; i > 0; i--) {<br>printf("%d ", i); if (i == 2) {<br>break;<br>}<br>}<br>return 0;<br>} | | | |
| | A) | 3 2 | B) | 3 2 1 |
| | C) | 3 2 1 0 | D) | 3 2 1 |
| **11.** | **Which of the following statements is true?** | | | |
| | A) | for loop is faster than while loop | B) | while loop is faster than for loop |
| | C) | do-while loop is faster than both for and while loops | D) | There is no difference in execution speed between for, while, and do-while loops |
| **12.** | Consider the following code snippet. What will be the value of 'x' after the loop finishes?<br><br>int x = 0;<br>for (int i = 0; i < 5; i++)<br>{<br>x += i;<br>} | | | |
| | A) | 5 | B) | 10 |
| | C) | 15 | D) | **20** |
| **13.** | **Which loop checks the condition after executing the loop body?** | | | |
| | A) | for loop | B) | while loop |
| | C) | do-while loop | D) | None of the above |
| **14.** | Which loop is most suitable when the number of iterations is known in advance? | | | |
| | A) | for loop | B) | while loop |
| | C) | do-while loop | D) | All of the above |
| **15.** | **What is the purpose of the else clause in an if-else statement?** | | | |

| | | | | |
|---|---|---|---|---|
| | A) | To check another condition | B) | To execute a block of code when the `if` condition is false |
| | C) | To end the program | D) | To repeat the `if` block |