

1) The correct order of compilation of a C program is

- i) Compile source code to object code
 - ii) Link object code and libraries
 - iii) Create binary or executable file
 - iv) Process Preprocessor directives
- A) i → ii → iii → iv
 - B) iv → i → ii → iii
 - C) i → iv → ii → iii

2) Which of the following data type can be used for storing a floating point constant?

- A) long int
- B) double
- C) unsigned char

3) What is the result of running the following code snippet?

```
float result;  
result = 5/2;  
printf("%.1f",result);
```

- A) 2.0
- B) 2.5
- C) 0

4) What is the result of running the following code snippet

```
if(0)  
    printf("The Sun rises in the East");  
else  
    printf("The Sun rises in the West");
```

- A) The Sun rises in the East
- B) The Sun rises in the West
- C) Compilation Error

5) What is the value of variable *i* after running the following statements?

```
int i = 10;  
i = i / 2;  
i = i % 2;
```

- A) 0
- B) 1
- C) 2

- 6) **The purpose of the function *int toupper (int c)* as defined in *ctype.h* is to**
- A) Convert the input string stored in variable *c* to upper case
 - B) Convert the input character stored in variable *c* to upper case
 - C) None of the above
- 7) **The difference between the user defined constants 'Z' and "Z" is**
- A) 'Z' is a string constant whereas "Z" is a character constant
 - B) 'Z' is a character constant whereas "Z" is a string constant
 - C) There is no difference
- 8) **Output of running the following code snippet is**
- ```
int i;
for (i = 0; i < 3; i++)
{
 if(i == 2)
 {
 printf("%d\t",i);
 }
}
```
- A) 0      1      2      3
  - B) 0      1      2
  - C) 2
- 9) **The conversion character for printing an integer in hexadecimal number using *printf* is**
- A) %x
  - B) %d
  - C) %o
- 10) **Output of following code snippet**
- ```
float marks[10] = {53,66,36,36,53,23,64,12,53,88};
float *marksPtr = marks;
marksPtr = marksPtr + 4;
printf("%.0f", *marksPtr);
```
- A) 0
 - B) 53
 - C) 36
 - D) Garbage value
- 11) **Which of the following is the correct way to allocate space for an array of 15 float variables?**
- A) `float *floatArray = (float *) malloc (15);`
 - B) `float floatArray = (float *) malloc (15);`
 - C) `float *floatArray = (float *) malloc (15 * sizeof(float));`
 - D) `float *floatArray = 15 * malloc (sizeof(float));`
- 12) **The following C statement causes an infinite loop (true/false):**

```
while (1);
```

- A) true
- B) false

13) Output of the following code snippet is:

```
int array[] = {1,2,3,4,5};  
printf("%d", array[5]);
```

- A) 0
- B) 4
- C) 5
- D) Garbage value

14) Output of following statement is

```
#define NUM 5  
int main()  
{  
    NUM++;  
    printf("%d", NUM);  
    return 0;  
}
```

- A) 5
- B) 6
- C) Compilation error

15) Output of the following statement is

```
int i = 5;  
if(i = 5)  
{  
    puts("Structured programming language");  
}  
else  
{  
    puts("Object oriented programming language");  
}
```

- A) Structured programming language
- B) Object oriented programming language
- C) Compilation error

- 16) Which of the following is a true statement?
- A) Strings in C are char arrays terminated by '\0'
 - B) Strings in C are char arrays terminated by NULL
 - C) Strings in C are char arrays terminated by EOF
- 17) Which of the following is a NOT a true statement?
- A) The base address of an array is same as the name of the array
 - B) The base address of an array is address of the first element in the array
 - C) The base address of a static array can be modified
- 18) Which of the following is true about break statement in C?
- A) break statement exits out of the innermost loop
 - B) break statement exits out of all the outer loops nested around it
 - C) break statement skips one iteration and continues looping
- 19) Output of the following program is:
- ```
int main()
{
 int i =0;
 modify(i);
 printf("%d", i);
 return 0;
}
int modify(int z)
{
 return (++z);
}
```
- A) 0
  - B) 1
- 20) Point out at least two errors in the following code snippet
- ```
float a;
scanf("Enter number: %f", a )
```