

## C-PROGRAMMING

1. Which of the following statements should be used to obtain a remainder after dividing 3.14 by 2.1 ?

- (a) `rem = 3.14 % 2.1;`
- (b) `rem = modf(3.14, 2.1);`
- (c) `rem = fmod(3.14, 2.1);`
- (d) Remainder cannot be obtained in floating point division.

2. What are the types of linkages?

- |                           |                                 |
|---------------------------|---------------------------------|
| (a) Internal and External | (b) External, Internal and None |
| (c) External and None     | (d) Internal                    |

3. Which of the following special symbol allowed in a variable name?

- |                  |                    |
|------------------|--------------------|
| (a) * (asterisk) | (b)   (pipeline)   |
| (c) - (hyphen)   | (d) _ (underscore) |

4. Is there any difference between following declarations?

- 1: `extern int fun();`
- 2: `int fun();`

- (a) Both are identical
- (b) No difference, except `extern int fun();` is probably in another file
- (c) `int fun();` is overridden with `extern int fun();`
- (d) None of these

5. How would you round off a value from 1.66 to 2.0?

- (a) ceil(1.66)
- (c) roundup(1.66)

- (b) floor(1.66)
- (d) roundto(1.66)

6. By default a real number is treated as a:

- (a) float
- (c) long double

- (b) double
- (d) far double

7. Which of the following is not user defined data type?

1: struct book

```
{  
    char name[10];  
    float price;  
    int pages;  
};
```

2: long int l = 2.35;

3: enum day {Sun, Mon, Tue, Wed};

- (a) 1
- (c) 3

- (b) 2
- (d) Both 1 and 2

8. Is the following statement a declaration or definition?

```
extern int i;
```

- (a) Declaration
- (c) Function

- (b) Definition
- (d) Error

9. Identify which of the following are declarations:

1: extern int x;

2: float square ( float x ) { ... }

3: double pow(double, double);

- (a) 1

- (b) 2

(c) 1 and 3

(d) 3

**10.** In the following program where is the variable a getting defined and where it is getting declared?

```
#include<stdio.h>
```

```
int main()
{
    extern int a;
    printf("%d\n", a);
    return 0;
}
int a=20;
```

- (a) extern int a is declaration, int a = 20 is the definition
- (b) int a = 20 is declaration, extern int a is the definition
- (c) int a = 20 is definition, a is not defined
- (d) a is declared, a is not defined

**11.** When we mention the prototype of a function?

- |                 |               |
|-----------------|---------------|
| (a) Defining    | (b) Declaring |
| (c) Prototyping | (d) Calling   |

**12.** What is (void\*)0?

- |                                    |                                    |
|------------------------------------|------------------------------------|
| (a) Representation of NULL pointer | (b) Representation of void pointer |
| (c) Error                          | (d) None of above                  |

**13.** Can you combine the following two statements into one?

```
char *p;
p = (char*) malloc(100);
```

- |  |   |
|--|---|
| (a) <code>char p = *malloc(100);</code>        | (b) <code>char *p = (char) malloc(100);</code>      |
| (c) <code>char *p = (char*)malloc(100);</code> | (d) <code>char *p = (char *) (malloc*)(100);</code> |

**14.** In which header file is the NULL macro defined?

- |  |                           |
|--|---------------------------|
| (a) <code>stdio.h</code>                           | (b) <code>stddef.h</code> |
| (c) <code>stdio.h</code> and <code>stddef.h</code> | (d) <code>math.h</code>   |

**15.** If a variable is a pointer to a structure, then which of the following operator is used to access data members of the structure through the pointer variable?

- |                    |                        |
|--------------------|------------------------|
| (a) <code>.</code> | (b) <code>&amp;</code> |
| (c) <code>*</code> | (d) <b>7</b>           |

**16.** What would be the equivalent pointer expression for referring the array element `a[i][j][k][l]`?

- |                                     |   |
|-------------------------------------|---|
| (a) <code>(((((a+i)+j)+k)+l)</code> | (b) <code>*(*(*(* (a+i)+j)+k)+l)</code> |
| (c) <code>((((a+i)+j)+k+l)</code>   | (d) <code>((a+i)+j+k+l)</code>          |

**17.** A pointer is:

- (a) A keyword used to create variables
- (b) A variable that stores address of an instruction
- (c) A variable that stores address of other variable
- (d) All of the above

**18.** The operator used to get value at address stored in a pointer variable is:

- |                             |                        |
|-----------------------------|------------------------|
| (a) <code>*</code>          | (b) <code>&amp;</code> |
| (c) <code>&amp;&amp;</code> | (d) <code>  </code>    |

**19.** What is the output of the program given below?

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    enum status { pass, fail, atkt};
```

```
    enum status stud1, stud2, stud3;
```

```
    stud1 = pass;
```

```
    stud2 = atkt;
```

```
    stud3 = fail;
```

```
    printf("%d, %d, %d\n", stud1, stud2, stud3);
```

```
    return 0;
```

```
}
```

(a) 0, 1, 2

(c) 0, 2, 1

(b) 1, 2, 3

(d) 1, 3, 2