C-PROGRAMMING

Solutions

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.

Solution: Option (c)

Explanation:
fmod(x,y) - Calculates x modulo y, the remainder of x/y.
This function is the same as the modulus operator. But fmod() performs floating point divisions.

2. What are the types of linkages?

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

Solution: Option (b)

Explanation:
External Linkage❼ means global, non-static variables and functions.
Internal Linkage❼ means static variables and functions with file scope.
None Linkage❼ means Local variables.

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

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

Solution: Option (d)

Explanation:
Variable names in C are made up of letters (upper and lower case) and digits. The underscore

character ("_") is also permitted. Names must not begin with a digit.

Examples of valid (but not very descriptive) C variable names:

- foo
- Bar
- BAZ
- foo_bar
- _foo42
- _
- QuUx

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 overrided with extern int fun();
(d) None of these

Solution: Option (b)

Explanation:
extern int fun(); declaration in C is to indicate the existence of a global function and it is defined externally to the current module or in another file.

int fun(); declaration in C is to indicate the existence of a function inside the current module or in the same file.

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

(a) ceil(1.66)
(b) floor(1.66)
(c) roundup(1.66)
(d) roundto(1.66)

Solution: Option (a)

Explanation:

```
/* Example for ceil() and floor() functions: */

#include<stdio.h>
#include<math.h>

int main()
{
printf("\n Result : %f" , ceil(1.44) );
printf("\n Result : %f" , ceil(1.66) );
printf("\n Result : %f" , floor(1.44) );
printf("\n Result : %f" , floor(1.66) );
return 0;
}

// Output:
// Result:  2.000000
// Result : 2.000000
// Result : 1.000000
// Result : 1.000000
```

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

(a) float
(b) double
(c) long double
(d) far double

Solution: Option (b)

Explanation:

In computing, 'real number' often refers to non-complex floating-point numbers. It includes both rational numbers, such as 42 and 3/4, and irrational numbers such as pi = 3.14159265.....

When the accuracy of the floating point number is insufficient, we can use the double to define the number. The double is same as float but with longer precision and takes double space (8 bytes) than float.

To extend the precision further we can use long double which occupies 10 bytes of memory space.

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 (b) 2
(c) 3 (d) Both 1 and 2

Solution: Option (b)

Explanation:
C data types classification are

1. Primary data types
   (a) int
   (b) char
   (c) float
   (d) double
   (e) void

2. Secondary data types (or) User-defined data types
   (a) Array
   (b) Pointer
   (c) Structure
   (d) Union
   (e) Enum

So, clearly long int l = 2.35; is not User-defined data type.
(i.e.long int l = 2.35; is the answer.)

8. Is the following statement a declaration or definition?
extern int i;

(a) Declaration
(b) Definition
(c) Function
(d) Error

Solution: Option (a)

Explanation:
Declaring is the way a programmer tells the compiler to expect a particular type, be it a variable, class/ struct/ union type, a function type (prototype) or a particular object instance. (ie. extern int i).

Declaration never reserves any space for the variable or instance in the program's memory; it simply a "hint" to the compiler that a use of the variable or instance is expected in the program. This hinting is technically called "forward reference".

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

Solution: Option (c)

Explanation:
extern int x; - is an external variable declaration.
double pow(double, double); - is a function prototype declaration.
Therefore, 1 and 3 are declarations. 2 is definition.

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

Solution: Option (a)

Explanation:

- ➢ During declaration we tell the datatype of the Variable.
- ➢ During definition the value is initialized.

11. When we mention the prototype of a function?

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

Solution: Option (b)

Explanation:
A function prototype in C or C++ is a declaration of a function that omits the function body but does specify the function's name, argument types and return type.

While a function definition specifies what a function does, a function prototype can be thought of as specifying its interface.

12. What is (void*)0?

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

Solution: Option (a)

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

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

(a) char p = *malloc(100);
(b) char *p = (char) malloc(100);
(c) char *p = (char*)malloc(100);
(d) char *p = (char *)(malloc*)(100);

Solution: Option (c)

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

(a) stdio.h
(b) stddef.h
(c) stdio.h and stddef.h
(d) math.h

Solution: Option (c)

Explanation:
The macro "NULL" is defined in locale.h, stddef.h, stdio.h, stdlib.h, string.h, time.h, and wchar.h.

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) .
(b) &
(c) *
(d) ❼

Solution: Option (d)

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

(a) ((((a+i)+j)+k)+l)
(b) *(*(*(*(a+i)+j)+k)+l)
(c) (((a+i)+j)+k+l)
(d) ((a+i)+j+k+l)

Solution: Option (b)

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

Solution: Option (c)

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

(a) *
(b) &
(c) &&
(d) ||

Solution: Option (a)

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
(b) 1, 2, 3
(c) 0, 2, 1
(d) 1, 3, 2

Solution: Option (c)

Explanation:

enum takes the format like {0,1,2..) so pass=0, fail=1, atkt=2

stud1 = pass (value is 0)
stud2 = atkt (value is 2)
stud3 = fail (value is 1)

Hence it prints 0, 2, 1.