



ST. LAWRENCE HIGH SCHOOL

A Jesuit Christian Minority Institution



WORKSHEET – 39

Topic : Pointers – calloc(), malloc(), pointer to array, call by value, call by reference

Subject: COMPUTER SCIENCE

Class - 12

F.M:15

Chapter: Pointers in C

Date: 9/11/2020

Choose the correct answer for each question:

15x1=15

1. Which is the correct syntax for malloc() in c?
 - a. ptr = (cast-type*) malloc(byte-size &)
 - b. ptr = (cast-type) malloc(byte-size *)
 - c. ptr = (cast-type*) malloc(byte-size)
 - d. All of these
2. How many bytes are allocated by the following statement?[Assume size of int is 4 bytes]

```
ptr = (int*) malloc(100 * sizeof(int));
```

 - a. 100
 - b. 4
 - c. 104
 - d. 400
3. Which is the correct syntax for calloc() in c?
 - a. ptr = (cast-type*)calloc(n, element-size);
 - b. ptr = (cast-type)calloc(n, element-size);
 - c. ptr = (cast-type*)calloc(n * element-size);
 - d. All of these
4. How many bytes are allocated by the following statement?[Assume size of int is 4 bytes]

```
ptr = (int*) calloc(25 , sizeof(int));
```

 - a. 100
 - b. 25
 - c. 125
 - d. 400
5. Choose the correct syntax for free method:
 - a. free (ptr)
 - b. free(ptr *)
 - c. free (*ptr)
 - d. free(&ptr)
6. Choose the correct syntax for realloc method:
 - a. realloc(ptr *, newSize)
 - b. realloc(ptr, newSize)
 - c. realloc(ptr &, oldSize)
 - d. realloc(ptr *, oldSize)
7. Which of the following is the correct syntax to send an array as a parameter to function?
 - a. func(&array);
 - b. func(#array);
 - c. func(*array);
 - d. func(array[size]);

8. Which of the following does not initialize ptr to null (assuming variable declaration of a as int a=0;)?
- a. int *ptr = &a;
 - b. int *ptr = &a – &a;
 - c. int *ptr = a – a;
 - d. All of the mentioned
9. Which of the following can never be sent by call-by-value?
- a) Variable
 - b) Array
 - c) Structures
 - d) Both Array and Structures

With reference to below code, answer the following questions (10-13):

```
#include <stdio.h>
void main()
{
    int a[3] = {1, 2, 3};
    int *p = a;
    printf("%p\t%p", p, a);
}
```

10. What will be the output of the above C code?
- a) Same address is printed
 - b) Different address is printed
 - c) Compile time error
 - d) Nothing
11. What value is hold by p[1] ?
- a. 1
 - b. 2
 - c. 3
 - d. Garbage value
12. What value is hold by a[2] ?
- a. 1
 - b. 2
 - c. 3
 - d. 0
13. What value is hold by pointer 'p' ?
- a. Address of a[3]
 - b. Address of a[2]
 - c. Address of a[1]
 - d. Address of a[0]
14. Correct syntax to declare pointer to pointer in C:
- a. int **ptr;
 - b. int *ptr*;
 - c. int &&ptr;
 - d. int *ptr;
15. Correct way to access the value of double pointer:
- a. *ptr
 - b. **ptr
 - c. &ptr
 - d. &&ptr

Phalguni Pramanik