Week 12 Assignment and Solution

1. Which of the following cannot be a structure member?
    a) Another structure
    b) Array
    c) Function
    d) None of the above

Solution: (c) Function

Function is not allowed to be a member of a structure member.

2. Which header file should be included to use functions like malloc() and calloc()?
    a) stdio.h
    b) math.h
    c) stdlib.h
    d) dos.h

Solution: (c) stdlib.h

3. In C which keyword to be used to give a datatype a new name
    a) typedef
    b) struct
    c) char
    d) none of the above

Solution: (a) typedef

The C programming language provides a keyword called typedef, which you can use to give a type a new name. You can use typedef to give a name to your user defined data types as well.

4. What is the output?

```
#include<stdio.h>
int main()
{
struct xyz{ int a;};
struct xyz obj1={1};
struct xyz obj2 = obj1;
printf("%d", obj2.a);
obj2.a = 100;
printf("%d", obj1.a);
printf("%d", obj2.a);
retutn 0;
}
```

Solution: 11100

5. What will be output?

```
#include <stdio.h>
int fun(int arr[]) {
  arr = arr+1;
  printf("%d ", arr[0]);
}
int main(void) {
  int arr[3] = {5, 10, 15};
```

```
    fun(arr);
    printf("%d ", arr[0]);
    printf("%d ", arr[1]);
    return 0;
}
```

a) 5 10 10

b) 10 5 15

c) 10 5 10

d) 10 15 5

Solution: (c) 10 5 10

In C, array parameters are treated as pointers So the variable arr represents an array in main(), but a pointer in fun().

6. What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires 4 bytes of memory

```
#include <stdio.h>
int main()
{
  unsigned int x[4][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};
  printf("%u, %u, %u", x+3, *(x+3), *(x+2)+3);
}
```

   a) 2036 2036 2036
   b) 2012 4 2204
   c) 2036 10 10
   d) 2012 4 6

   Solution: (a) 2036, 2036, 2036

x = 2000

Since x is considered as a pointer to an array of 3 integers and an integer takes 4 bytes, value of x + 3 = 2000 + 3*3*4 = 2036

The expression, *(x + 3) also prints same address as x is 2D array.
The expression *(x + 2) + 3 = 2000 + 2*3*4 + 3*4 = 2036

7. The program will allocate …….bytes to ptr. Assume sizeof(int)=4.

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

  int main()
  {
     int *ptr;
     ptr = (int*)malloc(sizeof(int)*4);
     ptr = realloc(ptr,sizeof(int)*2);
     return 0;
  }
```

Solution: 8

We can also use the realloc() to change memory block size.

8. What is the output?

```
#include<stdio.h>
#include<stdlib.h>
 int main()
  {
     int *ptr;
     ptr = (int *)calloc(1,sizeof(int));
     if (ptr != NULL)
         printf("%d\n",*ptr);
     return 0;
  }
```

   a) 0
   b) 1
   c) Garbage value
   d) Compilation error

Solution: (a)
The memory allocated by calloc() contains 0 until process does not make any change to it.

9. What is the output of the following C program?

```
#include <stdio.h>
main()
{
        int *p, a=10;
        p=&10;
        printf("%d",*p);
}
```

   a) 10
   b) a
   c) address of a
   d) compilation error

Solution: (d) A pointer variable can be assigned as the address of any constant. Thus, the compiler will show error as “[Error] lvalue required as unary '&' operand”.

10. Find the output of the C code given below

```
#include <stdio.h>
int main()
{
   char *s = "hello";
   char *p = s;
   printf("%p\t%p", p, s);
   return 0;
}
```

   a) Different address is printed
   b) Same address is printed
   c) Run time error
   d) Nothing

Solution: (b) Pointers s and p contain same address.

11. Find the output of the following C program.

```
#include <stdio.h>
fun(char *k)
```

```
{
    printf("%s", k);
}

int main()
{
    char s[] = "hello";
    fun(s);
    return 0;
}
```

a) hello
b) h
c) compilation error
d) No output

Solution: (a) The pointer k contains the starting address of the array s. Printing the string with starting address of s will print the content of the addresses till it finds the \0 character, Thus, hello will be printed.

12. Calling a function 'f' with a an array variable a[3] where 'a' is an array, is equivalent to
    a) f(a[3])
    b) f(*(a + 3))
    c) f(3[a])
    d) all the above methods are correct

Solution: (d) all the methods are correct.

13. What if the output of the following C program?

```
#include <stdio.h>
void m(int *p, int *q)
{
    int temp = *p; *p = *q; *q = temp;
}
int main()
{
    int a = 6, b = 5;
    m(&a, &b);
    printf("%d %d", a, b);
return 0;
}
```

Solution: 5 6 The program performs swapping of the variable a and b. So, the printed value is 5 6.

14. Find the output of the C program given below.

```
# include <stdio.h>
int main( )
{
    char s1[7] = "1234", *p;
    p = s1 + 2;
    *p = '0' ;
    printf ("%s", s1);
  return 0;
}
```

Solution: The 3rd element of s1 will be replaced with 0. Thus, the correct output is 1204.

15. Consider the following function written in the C programming language. The output of the above function on input “ABCD EFGH” is

```
#include <stdio.h>
void foo(char *a)
{
  if (*a && *a != ' ')
  {
     foo(a+1);
     putchar(*a);
  }
}
```

a) ABCD EFGH
b) ABCD
c) HGFE DCBA
d) DCBA

Solution: (d) The program prints all characters before ‘ ‘ or ‘\0’ (whichever comes first) in reverse order. Thus, the correct output is DCBA.