ASSIGNMENT-7 SOLUTION

1. The searching operation in an array is done using
   a) Key and index
   b) Only key
   c) Only index
   d) None of these

   Solution: (a) Both key and array index are used to perform search operation in arrays.

2. The last character of a string is
   a) \n
   b) \0
   c) \b
   d) \t

   Solution: (b) \0 is the indicator of the end of the string.

3. The right method of initializing a 2D array is
   a) int abc[2][2] = {1, 2, 3, 4}
   b) int abc[][] = {1, 2, 3 ,4}
   c) int abc[2][] = {1, 2, 3 ,4}
   d) all of the above

Solution: (a) The valid initialization is option (a). Next two are invalid declaration because the second dimension must be specified.

4. Applications of multidimensional array are?

   a) Matrix-Multiplication
   b) Minimum Spanning Tree
   c) Finding connectivity between nodes
   d) All of the mentioned

Solution: (d) For all of the above cases, multi-dimensional arrays are used.

5. In C, the placement of elements of a two dimensional array is
   a) Row wise
   b) Column wise
   c) Diagonal wise
   d) Bottom to top wise

Solution: (a) In C the placement of 2D array in memory is row wise.

6. If the starting address of an float array Arr[10][10] is 2000, what would be the memory address of the element Arr[5][6]? (float takes 4 bytes of memory)
   a) 2268
   b) 2120
   c) 2224
   d) 2144

Solution: (c) If ‘a’, ‘b’ and ‘c’ denotes the starting address, number of columns and size in bytes for each element respectively of array Arr[][], then the location of Arr[i][j] can be calculated as

$$Address = a + (i * b + j) * c$$

Thus the address of Arr[5][6] is 2000+(5*10+6)*4=2224

7. What will be the output of the code below?

```
#include <stdio.h>
int main()
{
   int disp[2][4] = {{5, 6, 8, 2}, {4, 5, 3, 7}};
   printf("%d\n", disp[1][0]);
   return 0;
}
```

Solution: 4

8. What is the output of the following C code?

```
#include <stdio.h>
int main()
{
 int ary[][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
 printf("%d\n", ary[2][0]);
 return 0;
}
```

a) Compile time error
b) 7
c) 1
d) 2

Solution: (b) In the initialization method of a multidimention array, it must have bounds for all dimensions except the first.

9. Find the output of the following C program.

```
#include <stdio.h>
int main()
{
char a[10][8] = {"hi", "hello", "fellows"};
printf("%s", a[2]);
return 0;
}
```

a) fellows
b) h
c) fello
d) Compiler error

Solution: (a) a[2] indicates the 3rd string of the 2D array. Thus “fellows” will be printed.

10. If the two strings s1 and s2 are identical, then strcmp(char *s1, char *s2) function returns

a) 1
b) -1
c) 0

d) Any Nonzero values

Solution: (c) strcmp (const char *s1, const char *s2);
The strcmp return an int value that is

if s1 < s2 returns a value < 0

if s1 == s2 returns 0

if s1 > s2 returns a value > 0

11. What will be the output?

```
# include <stdio.h>
int main()
 {
  char str1[] = "Week-7-Assignment";
  char str2[] = {'W', 'e', 'e', 'k', '-', '7', '-', 'A', 's','s','i','g','n','m','e','n','t'};
  int n1 = sizeof(str1)/sizeof(str1[0]);
  int n2 = sizeof(str2)/sizeof(str2[0]);
 printf("n1 = %d, n2 = %d", n1, n2);
 return 0;
}
```

a) n1=18, n2=17
b) n1=18, n2=18
c) n1=17, n1=17
d) n1=17, n2=18

Solution: (a) The size of str1 is 18 and size of str2 17.

When an array is initialized with string in double quotes, compiler adds a '\0' at the end.

12. Consider the following C program segment:

```
#include<stdio.h>
#include<string.h>
int main()
{
char p[20];
char s[] = "string";
int length = strlen(s);
int i;
for (i = 0; i < length; i++)
    p[i] = s[length - i];
printf("%s", p);
return 0;
}
```

The output would be:

a) gnirts

b) gnirt
c) string
d) no output is printed

Solution: (d)
Let us consider below line inside the for loop p[i] = s[length — i];
For i = 0, p[i] will be s[6 - 0] and s[6] is '\0′
So p[0] becomes '\0'. It doesn't matter what comes in p[1], p[2]….. as P[0] will not change for i >0. Nothing is printed if we print a string with first character '\0′

13. What will be the value of 'i' after the execution of the C code given below?

```
#include<stdio.h>
#include<string.h>
int main()
  {
   static char str1[] = "dills";
   static char str2[20];
   static char str3[] = "daffo";
   int i;
   i = strcmp(strcat(str3, strcpy(str2, str1)), "daffodills");
   return 0;
  }
```

a) 0
b) 1
c) -1
d) None

Solution: (a) 0
strcat(str3, strcpy(str2, str1)) makes it "daffodills", hence strcmp("daffodills", "daffodills")=0

14. What will be the output?

```
# include <stdio.h>
int main()
   {
      int a[2][3] = {1, 2, 3, 4};
      int i = 0, j = 0;
      for (i = 0; i < 2; i++)
      for (j = 2; j >=0; j--)
      printf("%d", a[i][j]);
      return 0;
   }
```

Solution: 321004

In a[2][3] = {1, 2, 3, 4}; only 4 values are given. The rest will be taken as 0. So, finally a[2][3] = {{1, 2, 3}, {4,0,0}}; So, 321004 will be printed as per the given for loop.

15. What will be the output?

```
#include<stdio.h>
int main()
  {
    int i;
    char a[] = "";
    if(printf("%s", a))
      printf("The string is empty");
    else
      printf("The string is not empty");
   return 0;
  }
```

a) The string is empty
b) The string is not empty
c) Error
d) None

Solution: (b) The string is not empty