**Labsheet 6**

Questions 26-30 Type, compile and execute the following programs and record the results and observations in the lab workbook. Get the signature from the faculties in charge of the lab towards the end of the lab

**Arrays**

1. Write a program which reads in 10 integers from the user and stores them in an array. Find the largest value in the array and print it.
2. Modify the last program to use a preprocessor constant for the size of the array and in the test condition of the loop which processes the array.
3. Modify the last program to find mean of *n* numbers using arrays.
4. Write a program to interchange the largest and the smallest number in the array.
5. Write a program to find the second biggest number using an array of *n* numbers.
6. Write a program to find whether the array of integers contain a duplicate number. If it’s there print the position of duplicate numbers.
7. Write a program to delete the duplicate numbers in an array.
8. Write a program which can store 10 integers in an array. Fill the array with “random” numbers using the library functions **rand()** instead of reading them from the user. Find the largest element in the array and print it out.

Each time **rand()** is called it returns a “random” integer. Use the mod operator ( % ) to get a value in the desired range. For example:

int **result;**

result = **rand() % 1000;**

will assign a random value in the range **0 – 999** to the variable **result.**

Make sure your program contains the line:

**#include <stdlib.h>** to include information about the **rand()** function.

1. Modify the last program so that instead of finding the largest element in the array, the program sorts the elements of the array into ascending order.
2. Write a program rotate that left rotates an array of size n by d elements.

Input 1 2 3 4 5 if d=2

Output should be 3 4 5 1 2

1. Write a program to insert an element into an array at a given position.
2. Write a program to find out whether a particular element is in the integer array using

Linear search.

1. Write a program to find out whether a particular element is in the integer array using

Binary search.

1. Write a program to sort an array of elements using Bubble sort.
2. Write a program to sort an array of elements using Selection sort.

**Strings**

**Reading Strings**

1. If we declare a string by writing char str[50]; Then str can be read from the user by using three ways:

1. Using scanf() function

2. Using gets() function

3. Using getchar() function repeatedly

Write program to read a string in the above three ways.

**Writing strings**

1. The string can be displayed on screen using three ways:

1. Using printf() function

2. Using puts() function

3. Using putchar()function repeatedly.

Modify the above program to display the string that you read.

1. Run the following program and analyze the result.

#include<stdio.h>

int main()

{

char str = “Hello”;

printf(“\n %s”,str);

printf(“\n %s”,&str);

printf(“\n%s”,&str[2]);

return 0;

}

1. Write a program to find the length of a string. Also use strlen( ) to do the same.
2. Write a program to copy on string to another without using any string library functions. Do the same operation using strcpy() function in string.h)
3. Write a program to convert characters of a string to upper case.

*(Note: ASCII code for A-Z varies from 65 to 91 and the ASCII code for a-z ranges from 97 to 123)*

1. Write a program to concatenate two strings. (Do the same operation using the string library function *strcat()* and analyze the behavior; you should include *string .h*)
2. Write a program to compare two strings. (Do the same operation using the string library function *strcmp()* and analyze the behavior; you should include *string .h*)
3. Write a program to check whether the entered string is a palindrome.
4. Write a program to check whether a substring is present in a given string.

Predict the output of the following:-

* 1. #include <stdio.h>

int main()

{

    int arr[5];

    arr++;

    printf("%u", arr);

    return 0;

}

* 1. int main()

{

  char arr[] = "Amrita School Of Engineering";

  printf("%d", sizeof(arr));

  return 0;

}

#include<stdio.h>

int main()

{

int a[5] = {5, 1, 15, 20, 25};

int i, j, m;

i = ++a[1];

j = a[1]++;

m = a[i++];

printf("%d, %d, %d", i, j, m);

return 0;

}

* 1. #include<stdio.h>

int main()

{

int arr[5], i=0;

while(i<5)

arr[i]=++i;

for(i=0; i<5; i++)

printf("%d, ", arr[i]);

return 0;

}

#include<stdio.h>

int main()

{

int arr[1]={10};

printf("%d\n", 0[arr]);

return 0;

}

#include<stdio.h>

int main()

{

float arr[] = {12.4, 2.3, 4.5, 6.7};

printf("%d\n", sizeof(arr)/sizeof(arr[0]));

return 0;

}

int main()

{

    int i;

    int arr[5] = {1};

    for (i = 0; i < 5; i++)

        printf("%d ", arr[i]);

    return 0;

}

#include <stdio.h>

int main()

{

char p;

char buf[10] = {1, 2, 3, 4, 5, 6, 9, 8};

p = (buf + 1)[5];

printf("%d\n", p);

return 0;

}

#include <stdio.h>

int size = 4;

int arr[size];

int main()

{

if(arr[0])

printf("Not Initialized to ZERO");

else

printf("initialized to ZERO");

return 0;

}

#include <stdio.h>

int arr[4];

int main()

{

if(arr[0])

printf("Not Initialized to ZERO");

else

printf("initialized to ZERO");

return 0;

}

#include <stdio.h>

int main ()

{

    int sum = 0, maxsum = 0,  i,  n = 6;

    int a [] = {2, -2, -1, 3, 4, 2};

    for (i = 0; i < n; i++)    {

            if (i == 0 || a [i]  < 0  || a [i] < a [i - 1])  {

                     if (sum > maxsum) maxsum = sum;

                     sum = (a [i] > 0) ? a [i] : 0;

            }

            else sum += a [i];

    }

    if (sum > maxsum) maxsum = sum ;

    printf ("%d\n", maxsum);

}

#include <stdio.h>

int main ()

{

char a [6] = "world";

int i, j;

for (i = 0, j = 5; i < j; a [i++] = a [j--]);

printf ("%s\n", a);

}

#include <stdio.h>

int main ()

{

int i, j;

int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};

for(i = 0; i < 3; i++) {

a[i] = a[i] + 1;

i++;

}

i--;

for (j = 7; j > 4; j--) {

int i = j/2;

a[i] = a[i] - 1;

}

printf ("%d, %d", i, a[i]);

}

\*\*\*\*\*\*\*\*\*\*\*\*\*