DAY-2

1.write a c program to initialize array and print the array?

Program:

```
#include <stdio.h>

int main (){
        int size ,i;
        printf("Enter the size of the array: ");
        scanf("%d",&size);
        int array[size];
        printf("Enter %d elements of array: \n", size);
        for (i = 0; i < size; i++){
                scanf("%d", &array[i]);
        }
        printf("Elements in array are :");
        for (i = 0;i < size; i++){
                printf("%d", array[i]);
        }
        printf("\n");
}
```

Output:

```
Enter the size of the array: 5
Enter 5 elements of array:
1
22
44
56
32
Elements in array are :122445632

--------------------------------
Process exited after 7.995 seconds with return value 10
Press any key to continue . . .
```

2.write a c program to find sum of elements in the given array?

Program:

```
#include <stdio.h>

int main ()
```

```
{
	int size ,i;
	printf("Enter the size of the array: ");
	scanf("%d",&size);
	int array[size];
	printf("enter the elements : ");
	for(i = 0; i < size; i++){
		scanf("%d", &array[i]);
	}
	int sum=0;
	for(i =0;i < size;i++){
		sum += array[i];
	}
	printf("The sum of elements in array : %d\n",sum);
	return 0;
}
```

Output:

```
Enter the size of the array: 5
enter the elements : 1
6
9
4
5
The sum of elements in array : 25

--------------------------------
Process exited after 8.122 seconds with return value 0
Press any key to continue . . .
```

3.write a c program to find sum of even and sum of odd numbers in an array?

Program:

```
#include <stdio.h>
int main ()
{
	int size ,i;
	printf("Enter the size of the array: ");
	scanf("%d",&size);
```

```
    int array[size];
    printf("enter the elements : ");
    for(i = 0; i < size; i++){
        scanf("%d", &array[i]);
    }
    int sumEven=0;
    int sumOdd=0;
    for(i =0;i < size;i++){
        if(array[i] %2==0){
            sumEven += array[i];
        }
        else{
            sumOdd += array [i];
        }
    }
    printf("The sum of even numbers in an array : %d\n",sumEven);
    printf("The sum of odd numbers in an array : %d\n",sumOdd);
    return 0;
}
```

Output:

```
Enter the size of the array: 5
enter the elements : 1
2
3
4
5
The sum of even numbers in an array : 6
The sum of odd numbers in an array : 9

--------------------------------
Process exited after 6.94 seconds with return value 0
Press any key to continue . . .
```

4.write a c program to merge the two array of elements?

Program:

```
#include <stdio.h>
int main ()
{
```

```
int size1,size2,i;
        printf("Enter the size of first array: ");
scanf("%d",&size1);
int array1[size1];
printf("Enter the elements of first array : ");
for(i = 0;i < size1;i++){
        scanf("%d",array1[i]);
}
printf("Enter the size of second array: ");
scanf("%d",&size2);
int array2[size2];
printf("Enter the elements of second array : ");
for(i = 0;i < size2;i++){
        scanf("%d",array2[i]);
}
int mergedsize = size1+size2;
int mergedarray[mergedsize];
for(i = 0;i < size1; i++){
        mergedarray[i]=array1[i];
}
for(i = 0;i < size2; i++){
        mergedarray[size1 + i]=array2[i];
}
printf("merged elements are : ");
for(i = 0;i < mergedsize;i++){
        printf("%d",mergedarray[i]);
}
printf("\n");
return 0;
}
```

Output:

```
Enter the size of the first array: 5
Enter the elements of the first array: 1
2
3
4
5
Enter the size of the second array: 5
Enter the elements of the second array: 10
9
8
7
6
Merged elements are: 1 2 3 4 5 10 9 8 7 6

--------------------------------
Process exited after 17.43 seconds with return value 0
Press any key to continue . . .
```

5.Write a c program to find duplicate element in an array?

Program:

```
#include <stdio.h>

int main ()

{
        int size ,i ,j;
        printf("Enter the size of an array: ");
        scanf("%d",&size);
        int array[size];
        printf("Enter the elements of array: \n",size);
        for(i = 0;i < size;i++){
                scanf("%d",&array[i]);
        }
        printf("Duplicate elements:");
        for(i = 0;i < size; i++){
                for(j = i + 1;j < size;j++){
                        if(array[i] == array[j]){
                                printf("%d",array[i]);
                                break;
                        }
                }
        }
```

```
    printf("\n");
    return 0;
}
```

Output:

```
Enter the size of an array: 5
Enter the elements of array:
1
2
3
3

5
Duplicate elements:3

--------------------------------
Process exited after 9.41 seconds with return value 0
Press any key to continue . . .
```

6.Write a c program to find greatest element in an array?

Program:

```
#include <stdio.h>
int main ()
{
    int size ,i ,j;
    printf("Enter the size of an array: ");
    scanf("%d",&size);
    int array[size];
    printf("Enter the elements of array: \n",size);
    for(i = 0;i < size;i++){
        scanf("%d",&array[i]);
    }
    int max = array[0];
    for(i = 0;i < size;i++){
        if(array[i] > max){
            max = array[i];
        }
    }
        printf("The greatest element in an array is : %d\n",max);
```

```
    return 0;
}
```

Output:

```
Enter the size of an array: 5
Enter the elements of array:
1
78
95
45
62
The greatest element in an array is : 95

--------------------------------
Process exited after 14.48 seconds with return value 0
Press any key to continue . . .
```

7.Write a c program to find element in an array using linear search?

Program:

```
#include <stdio.h>
int main() {
   int size, key ,i;
   printf("Enter the size of the array: ");
   scanf("%d", &size);
   int array[size];
   printf("Enter %d elements:\n", size);
   for ( i = 0; i < size; i++) {
      scanf("%d", &array[i]);
   }
   printf("Enter the element to search: ");
   scanf("%d", &key);
   int found = 0;
   int index = -1;
   for ( i = 0; i < size; i++) {
      if (array[i] == key) {
         found = 1;
         index = i;
         break;
      }
```

```
    }
    if (found) {
        printf("Element %d found at index %d.\n", key, index);
    } else {
        printf("Element %d not found in the array.\n", key);
    }
    return 0;
}
```

Output:

```
Enter the size of the array: 5
Enter 5 elements:
1
2
3
4
5
Enter the element to search: 4
Element 4 found at index 3.

--------------------------------
Process exited after 11.74 seconds with return value 0
Press any key to continue . . .
```

8.Write a c program to find element in an array using binary search?

Program:

```
#include <stdio.h>
int binarySearch(int array[], int size, int key) {
    int left = 0;
    int right = size - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (array[mid] == key) {
            return mid;
        } else if (array[mid] < key) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
```

```
    }
    return -1;
}
int main() {
    int size, key ,i;
    printf("Enter the size of the sorted array: ");
    scanf("%d", &size);
    int array[size];
    printf("Enter %d elements in sorted order:\n", size);
    for ( i = 0; i < size; i++) {
        scanf("%d", &array[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &key);
    int index = binarySearch(array, size, key);
    if (index != -1) {
        printf("Element %d found at index %d.\n", key, index);
    } else {
        printf("Element %d not found in the array.\n", key);
    }
    return 0;
}
```

Output:

```
Enter the size of the sorted array: 5
Enter 5 elements in sorted order:
1
2
3
4
5
Enter the element to search: 6
Element 6 not found in the array.

--------------------------------
Process exited after 8.509 seconds with return value 0
Press any key to continue . . .
```

9.Write a c program to reverse a given String?

Program:

```
#include <stdio.h>
#include <string.h>
void reverseString(char str[]) {
   int length = strlen(str) ,i;
   for ( i = 0; i < length / 2; i++) {
      char temp = str[i];
      str[i] = str[length - 1 - i];
      str[length - 1 - i] = temp;
   }
}
int main() {
   char input[100];
   printf("Enter a string: ");
   scanf("%s", input);
   reverseString(input);
   printf("Reversed string: %s\n", input);
   return 0;
}
```

Output:

```
Enter a string: 123456
Reversed string: 654321

--------------------------------
Process exited after 9.49 seconds with return value 0
Press any key to continue . . .
```

10.Write a c program to find string is palindrome or not?

Program:

```
#include <stdio.h>
#include <string.h>
int isPalindrome(char str[]) {
   int length = strlen(str) ,i;
   for ( i = 0; i < length / 2; i++) {
```

```
        if (str[i] != str[length - 1 - i]) {
            return 0;
        }
    }
    return 1;
}
int main() {
    char input[100];
    printf("Enter a string: ");
    scanf("%s", input);
    if (isPalindrome(input)) {
        printf("%s is a palindrome.\n", input);
    } else {
        printf("%s is not a palindrome.\n", input);
    }
    return 0;
}
```

Output:

```
Enter a string: malayalam
malayalam is a palindrome.

--------------------------------
Process exited after 6.977 seconds with return value 0
Press any key to continue . . .
```

11.write a c program to find and count number of times vowels are present in given string?

Program:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int isVowel(char ch) {
    ch = tolower(ch);
    return (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u');
}
```

```
int main() {
    int i;
    char input[100];
    printf("Enter a string: ");
    scanf("%s", input);
    int vowelCount[5] = {0};
    for (i = 0; i < strlen(input); i++) {
        if (isVowel(input[i])) {
            switch (tolower(input[i])) {
                case 'a':
                    vowelCount[0]++;
                    break;
                case 'e':
                    vowelCount[1]++;
                    break;
                case 'i':
                    vowelCount[2]++;
                    break;
                case 'o':
                    vowelCount[3]++;
                    break;
                case 'u':
                    vowelCount[4]++;
                    break;
            }
        }
    }
    printf("Number of vowels in the string: %d\n", vowelCount[0] + vowelCount[1] + vowelCount[2] +
vowelCount[3] + vowelCount[4]);
    printf("Number of 'a' vowels: %d\n", vowelCount[0]);
```

```
    printf("Number of 'e' vowels: %d\n", vowelCount[1]);
    printf("Number of 'i' vowels: %d\n", vowelCount[2]);
    printf("Number of 'o' vowels: %d\n", vowelCount[3]);
    printf("Number of 'u' vowels: %d\n", vowelCount[4]);
    return 0;
}
```

Output:

```
Enter a string: saveetha
Number of vowels in the string: 4
Number of 'a' vowels: 2
Number of 'e' vowels: 2
Number of 'i' vowels: 0
Number of 'o' vowels: 0
Number of 'u' vowels: 0

--------------------------------
Process exited after 16.54 seconds with return value 0
Press any key to continue . . .
```

12. write a c program for matrix multiplication?

Program:

```
#include<stdio.h>
int main() {
    int a[10][10], b[10][10], c[10][10], n, i, j, k;
    printf("Enter the value of N (N <= 10): ");
    scanf("%d", & n);
    printf("Enter the elements of Matrix-A: \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", & a[i][j]);
        }
    }
    printf("Enter the elements of Matrix-B: \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", & b[i][j]);
        }
```

```
    }
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            c[i][j] = 0;
            for (k = 0; k < n; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    printf("The product of the two matrices is: \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("%d\t", c[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Enter the value of N (N <= 10): 3
Enter the elements of Matrix-A:
1 2 1
3 5 4
 1 6
 9
Enter the elements of Matrix-B:
3 3 3
3 3 3
3 3 3
The product of the two matrices is:
12      12      12
36      36      36
48      48      48

--------------------------------
Process exited after 26.89 seconds with return value 0
Press any key to continue . . .
```

13.Write a c program to perform following operations into an array 1)Insert an element 2)delete an element?

Program:

```c
#include <stdio.h>
#define MAX_SIZE 100
void displayArray(int arr[], int size) {
    printf("Array:");
    for (int i = 0; i < size; i++) printf(" %d", arr[i]);
    printf("\n");
}
void insertElement(int arr[], int *size, int position, int element) {
    if (*size >= MAX_SIZE || position < 0 || position > *size) {
        printf("Invalid operation!\n");
        return;
    }
    for (int i = *size; i > position; i--) arr[i] = arr[i - 1];
    arr[position] = element;
    (*size)++;
    displayArray(arr, *size);
}
void deleteElement(int arr[], int *size, int position) {
    if (*size <= 0 || position < 0 || position >= *size) {
        printf("Invalid operation!\n");
        return;
    }
    for (int i = position; i < *size - 1; i++) arr[i] = arr[i + 1];
    (*size)--;
    displayArray(arr, *size);
}
int main() {
    int arr[MAX_SIZE], size;
    printf("Enter initial size of the array: ");
    scanf("%d", &size);
```

```
    if (size < 0 || size > MAX_SIZE) {
        printf("Invalid size!\n");
        return 1;
    }
    printf("Enter %d elements for the array:\n", size);
    for (int i = 0; i < size; i++) scanf("%d", &arr[i]);
    printf("\nArray initially: ");
    displayArray(arr, size);
    int choice, element, position;
    printf("\nMenu:\n1. Insert an element\n2. Delete an element\nEnter your choice: ");
    scanf("%d", &choice);
    switch (choice) {
        case 1:
            printf("Enter the element to insert and its position: ");
            scanf("%d %d", &element, &position);
            insertElement(arr, &size, position, element);
            break;
        case 2:
            printf("Enter the position to delete: ");
            scanf("%d", &position);
            deleteElement(arr, &size, position);
            break;
        default:
            printf("Invalid choice!\n");
    }
    return 0;
}
```

Output:

```
Enter initial size of the array: 3
Enter 3 elements for the array:
1
2
3
Array initially: Array: 1 2 3

Menu:
1. Insert an element
2. Delete an element
Enter your choice: 1
Enter the element to insert and its position: 53
3
Array: 1 2 3 53
```