

Assignment No.8

Q1. Find minimum and maximum number in array.

-:Code :-

```
//Create all array using malloc in assign 8
// Find minimum and maximum number in array

#include<stdio.h>
#include<stdlib.h>
int greatest(int*,int);
int smallest(int*,int);
void main() {
    int n,i,great,small;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr=(int*) malloc(sizeof(int)*n);
    printf("Enter the Array :");
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    great=greatest(arr,n);
    small=smallest(arr,n);
    for(i=0;i<n;i++){
        if(arr[i]<small){
            small=arr[i];
        }
    }
    printf("Smallest number in array is :%d \n",small);
    printf("Greatest number in array is :%d \n",great);
    free(arr);
}

int greatest(int* arr,int n){
    int i,great=arr[0];
    for(i=0;i<n;i++){
        if(arr[i]>great){
            great=arr[i];
        }
    }
}
```

```
        return great;
    }
    int smallest(int* arr,int n){
        int i,small=arr[0];
        for(i=0;i<n;i++){

            if(arr[i]<small){
                small=arr[i];
            }
        }
        return small;
    }
```

Q2. Search the given number in array.

-:Code :-

```
//Search the given number in array.
```

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

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

```
void main(){
```

```
    int i,n;
```

```
    printf("Enter the Size of array :");
```

```
    scanf("%d",&n);
```

```
    int* arr=(int*) malloc(sizeof(int)*n);;
```

```
    for(i=0;i<n;i++){
```

```
        scanf("%d",&arr[i]);
```

```
    }
```

```
    i=searchnum(arr,n);
```

```
    if(i!=0){
```

```
        printf("The given search number is present on %d position in array
```

```
\n",i);
```

```
    }
```

```
    else{
```

```
        printf("The number is not prasant in Array");
```

```
    }
```

```
    free(arr);
```

```
}
```

```
int searchnum(int* arr,int n){
```

```
    int i,search,check=0;
```

```
    printf("Enter the Number you want to search :");
```

```
    scanf("%d",&search);
```

```
    for(i=0;i<n;i++){
```

```
        if(arr[i]==search){
```

```
            return i+1;
```

```
        }
```

```
    }
```

```
    if(check==0){
```

```
        return 0;
```

```
    }
```

```
}
```

Q3. Find sum of all numbers.

-:Code :-

```
//Find sum of all numbers.
```

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

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

```
int sumofnum(int*,int);
```

```
void main(){
```

```
    int n,i;
```

```
    printf("Enter the Size of array :");
```

```
    scanf("%d",&n);
```

```
    int* arr=(int*) malloc(sizeof(int)*n);
```

```
    for(i=0;i<n;i++){
```

```
        scanf("%d",&arr[i]);
```

```
    }
```

```
    int sum=sumofnum(arr,n);
```

```
        printf("The Sum of numbers in Array is : %d",sum);
```

```
    free(arr);
```

```
}
```

```
int sumofnum(int* arr,int n){
```

```
    int i,sum=0;
```

```
    for(i=0;i<n;i++){
```

```
        sum=sum+arr[i];
```

```
    }
```

```
    return sum;
```

```
}
```

```
}
```

Q4. Find odd and even among the numbers.

-:Code :-

//Find odd and even among the numbers.

```
#include<stdio.h>
#include<stdlib.h>
void evenum(int*,int);
void main(){
    int n,i;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr=(int*) malloc(sizeof(int)*n);
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    evenum(arr,n);
    free(arr);
}
void evenum(int* arr,int n){
    int i,j=0,k=0;
    int even[n],odd[n];
    for(i=0;i<n;i++){
        if(arr[i]%2==0){
            even[j]=arr[i];
            j++;
        }else{
            odd[k]=arr[i];
            k++;
        }
    }
    printf("\n The Even numbers in Array is : ");
    for(i=0;i<j;i++){
        printf(" %d",even[i]);
    }

    printf("\n The Odd numbers in Array is :");
    for(i=0;i<k;i++){
        printf(" %d",odd[i]);
    }
}
```

Q5. Print alternate elements in array.

-:Code :-

```
//Print alternate elements in array

#include<stdio.h>
#include<stdlib.h>
void alternum(int*,int);
void main(){
    int n,i;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr=(int*) malloc(sizeof(int)*n);
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }

    alternum(arr,n);

    free(arr);

}
void alternum(int* arr,int n){
    int i;
    for(i=0;i<n;i++){
        if(arr[i]%2==0){
            printf(" %d",arr[i]);
        }
    }
    for(i=0;i<n;i++){
        if(arr[i]%2!=0){
            printf(" %d",arr[i]);
        }
    }
}
```

Q6. Accept array and print only prime numbers of array

-:Code :-

//Accept array and print only prime numbers of array.

```
#include<stdio.h>
#include<stdlib.h>
void alternum(int* arr,int n);
void main(){
    int n,i;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr=(int*) malloc(sizeof(int)*n);
    int* even=(int*) malloc(sizeof(int)*n);
    int* odd==(int*) malloc(sizeof(int)*n);;
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    alternum(arr,n);
    free(arr);
    free(even);
    free(odd);
}
void alternum(int* arr,int n){
    int i;
    for(i=1;i<n;i++){
        if(arr[i]==2||arr[i]==3||arr[i]==5||arr[i]==7){
            printf(" %d",arr[i]);
        }
        if(arr[i]%2!=0&&arr[i]%3!=0&&arr[i]%5!=0&&arr[i]%7!=0){
            printf(" %d",arr[i]);
        }
    }
}
```

Q7. Take two array and add sum in third array

-:Code :-

//Take two array and add sum in third array

```
#include<stdio.h>
#include<stdlib.h>
void sumofarray(int*,int*,int);
void main(){
    int n,i;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr1=(int*) malloc(sizeof(int)*n);
    int* arr2=(int*) malloc(sizeof(int)*n);
    printf("Enter the First Array:\n");
    for(i=0;i<n;i++){
        scanf("%d",&arr1[i]);
    }
    printf("Enter the Second Array:\n");
    for(i=0;i<n;i++){
        scanf("%d",&arr2[i]);
    }
    sumofarray(arr1,arr2,n);
    free(arr1);
    free(arr2);
}
void sumofarray(int* arr1,int* arr2,int n){
    int i;
    printf("Sum of Both Array is:\n");
    for(i=0;i<n;i++){
        printf(" %d",arr1[i]+arr2[i]);
    }
}
```


Q8. Merge two arrays

-:Code :-

// Merge two arrays

```
#include<stdio.h>
#include<stdlib.h>
int* meargeofarray(int*,int*,int*,int,int,int);
void main(){
    int n1,n2,i;
    printf("Enter the Size of array :");
    scanf("%d",&n1);
    int* arr1=(int*) malloc(sizeof(int)*n1);
    printf("Enter the First Array:\n");
    for(i=0;i<n1;i++){
        scanf("%d",&arr1[i]);
    }
    printf("Enter the Size of array :");
    scanf("%d",&n2);
    int* arr2=(int*) malloc(sizeof(int)*n2);
    printf("Enter the Second Array:\n");
    for(i=0;i<n2;i++){
        scanf("%d",&arr2[i]);
    }
    int n=n1+n2;
    int* arr=(int*) malloc(sizeof(int)*n);
    meargeofarray(arr,arr1,arr2,n,n1,n2);
    printf("Merge of Both Array is:\n");
    for(i=0;i<n;i++){
        printf(" %d",arr[i]);

        free(arr);
        free(arr1);
        free(arr2);
    }

}

int* meargeofarray(int* arr,int* arr1,int* arr2,int n,int n1,int n2){
    int i,j;
    for(i=0;i<n1;i++){
```

```
        arr[i]=arr1[i];
    }
    for(j=0;j<n;j++,i++){
        arr[i]=arr2[j];
    }
    return arr;
}
```

Q9. Reverse the given array.

-:Code :-

//Reverse the given array

```
#include<stdio.h>
#include<stdlib.h>
void main(){
    int n,i;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr=(int*) malloc(sizeof(int)*n);
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }

    for(i=n-1;i>=0;i--){

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

    }
    free(arr);
}
```

Q10. Sort the array.

-:Code :-

//Sort the array

```
#include<stdio.h>
#include<stdlib.h>
void sortarray(int*,int);
void main(){
    int n,i;
    printf("Enter the Size of array :");
    scanf("%d",&n);
    int* arr=(int*) malloc(sizeof(int)*n);
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }

    sortarray(arr,n);
    for(i=0;i<n;i++){
        printf(" %d",arr[i]);
    }
    free(arr);
}
void sortarray(int* arr,int n){
    int i,j,temp;
    for(i=0;i<n-1;i++){
        for(j=i+1;j<n;j++){
            if(arr[i]>arr[j]){
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
```