S. Tulasi
AP19110010452
CSE-F

Assignment-6

1) i) Binary search by sorting array where element is given by user.

```
#include <stdio.h>
int n;
int A[5];
void sort();
void binary_search(int k);
int main(){
    int i, a;
    printf("Enter the number of elements of array");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("Enter the element");
        scanf("%d", &A[i]);
    }
    printf("Enter the element to search");
    scanf("%d", &a);
    sort();
    binary_search(a);
    return 0;
}
void sort(){
    int i, c, d, pos, temp;
    for(c=0; c<n; c++){
        pos=c;
        for(d=c+1; d<5; d++){
```

```
        if (A[pos] < A[d]){
            pos = d;
        }
    }
    if (pos != c) {
        temp = A[c];
        A[c] = A[pos];
        A[pos] = temp;
    }
}
printf("SORTED ARRAY");
for(i=0; i<n; i++) {
    printf ("%d /t", A[i]);
}
void binary_search (int k) {
    int fir = 0;
    int las = n-1;
    int mid = (fir+las)/2;
    while (fir <= las) {
        if (A[mid] < k) {
            fir = mid+1;
        }
        else if (A[mid] == k) {
            printf ("\n %d is found at %d", k, mid+1);
            break;
```

```
    else{
        las=mid-1;
    }
    mid=(las+fir)/2
  }
  if(fir>las){
      printf("\n not found in array");
  }
}
```

b) Ask user for 2 locations and print sum and product of numbers in the location.

```
#include<stdio.h>
int n;
int A[10];
void sort();
void operation(int a, int b);
int main(){
   int i,a,b;
   printf("Enter the number of elements in array");
   scanf("%d",&n);
   for(i=0;i<n;i++){
      printf("Enter the element");
      scanf("%d",&A[i]);
   }
   printf("Enter the positions to operate");
   scanf("%d%d",&a,&b);
   sort();
```

```
    operation(a,b);
    return 0;
}
void sort(){
    int i, c, d, pos, temp;
    for(c=0; c<n; c++){
        pos=c;
        for(d=c+1; d<n; d++){
            if (A[pos]>A[d]){
                pos=d;
            }
        }
        if (pos!=c){
            temp=A[c];
            A[c]=A[pos];
            A[pos]=temp;
        }
    }
    printf("SORTED ARRAY");
    for(i=0;i<n;i++){
        printf("%d\t", A[i]);
    }
}
void operation(int a, int b){
    int sum, product;
    sum=A[a]+A[b];
```

```
    product = A[a] * A[b];
    printf ("sum = %d", sum);
    printf ("product = %d", product);
}
```

2) Sort an array using merge sort and ask user to enter a number find product from first to number and number to last.

```
#include <stdio.h>
int n;
int A[10];
void product (int k);
void merge_sort (int i, int j);
void merge (int i1, int j1, int i2, int j2);
int main (){
  int i, k;
  printf ("Enter the number of elements in array");
  scanf ("%d", &n);
  for(i=0; i<n; i++){
     Printf ("Enter the element ");
     scanf ("%d", &A[i]);
  }
  printf ("Enter the position");
  scanf ("%d", &k);
  merge_sort (0, n-1);
  product (k)
  return 0;
}
```

```
void merge_sort(int i, int j){
    int mid;
    if (i<j){
        mid = (i+j)/2;
        merge_sort(i, mid);
        merge_sort(mid+1, j);
        merge(i, mid, mid+1, j);
    }
}
void merge(int i1, int j1, int i2, int j2){
    int temp[100];
    int i, j, k;
    i = i1;
    j = j1;
    k = 0;
    while(i<=j1 && j<=j2){
        if(A[i]<A[j]){
            temp[k++] = A[i++];
        }
        else{
            temp[k++] = A[i++];
        }
    }
    while(i<=j1){
        temp[k++] = A[i++];
    }
    while(j<=j2){
```

```
    temp[k++]=A[j++];
  }
  for(i=l1, j=0; i<=j2; i++,j++){
      A[i]=temp[j];
  }
}
void product(int k){
    int pro1=1, pro2=1, i;
    for(i=0; i<k; i++){
      pro1= pro1* A[i];
    }
    for(i=n-1; i>=k; i--){
      pro2 = pro2* A[i];
    }
    printf("product1= %d", pro1);
    printf("product2 = %d", pro2);
}
```

3) Insertion sort :-

Repeatedly swap i with the one to its left if smaller.

14	33	27	10

compaaress 14 and 33 (already in order) moves ahead (compaares 27 and 33) not in order so swaps.

14	27	33	10

compares 14 and 27 (already inorder)
moves ahead (compares 33 and 10)
not in order swaps -

14	27	10	33

compares 27 and 10. not in order so swap.

14	10	27	33

compares 14 and 10. not in order so swaps

10	14	27	33

(SORTED ARRAY).

Selection:- Directly swaps first element with smallest value, and second with second smallest and so on untill array is sorted.

14	33	27	10

smallest value 10 swaps 14, 10.

10	33	27	14

second smallest is 14 swaps 33 and 14.

10	14	27	33

(SORTED ARRAY).

4) Sort array using bubble sort
1) print in alternative order.

```
#include <stdio.h>
int n;
int A[10];
void sort();
void alternative();
int main(){
   int i;
   printf("Enter the number of elements");
   scanf("%d",&n);
   for(i=0;i<n;i++){
     printf("Enter element");
     scanf("%d",&A[i]);
   }
   sort();
   alternative();
   return 0;
}
void sort(){
   int i,j,temp;
   for(i=0;i<n;i++){
      for(j=0;j<n-i-1;j++){
         if(A[j]>A[j+1]){
```

```
            temp=A[j];
            A[j]=A[j+1];
            A[j+1]=temp;
        }
    }
}
    printf("SORTED ARRAY");
    for(i=0;i<n;i++){
        printf("%d/t", A[i]);
    }
}
void alternative(){
    int i=0;
    while(i<5){
        printf("\n%d\t",A[i]);
        i=i+2;
    }
}
```

ii) sum of elements in odd pos and pro in even po

```
#include <stdio.h>
int n;
int A[10];
void sort();
void sum();
void product();
int main(){
```

```
int i;
printf ("Enter the number of elements");
scanf ("%d", &n);
for(i=0; i<n; i++) {
    printf ("Enter element");
    scanf ("%d", &A[i]);
}
sort();
sum();
product();
return 0;
}
void sort(){
  int i, j, temp;
  for(i=0; i<n; i++){
    if (A[j] > A[j+1]) {
      temp= A[j];
       A[j]= A[j+1];
      A[j+1] = temp;
    }
  }
  printf ("SORTED ARRAY");
  for(i=0; i<n; i++){
      printf ("%d/t", A[i]);
  }
}
```

```
void sum() {
    int i, sum=0;
    while (i<5) {
        sum = sum + A[i];
        i = i + 2;
    }
    printf ("sum=%d", sum);
}
void product ( ) {
    int i=0;
    int product=1;
    while (i<5) {
        product = product * A[i];
        i = i + 2;
    }
    printf ("product=%d", product);
}
```

iii) To print number divisible by m in array where m is given by user.

```
#include <stdio.h>
int n;
int A[10];
void sort();
void divisible (int m);
int main() {
```

```
    int i,m;
    printf("Enter the number of elements in array");
    scanf("%d", &n);
    for(i=0; i<n; i++){
      printf("Enter the element ");
      scanf("%d", &A[i]);
    }print("Enter the number");
    sort();      scanf("%d",&m);
    divisible();
}   return 0;

void sort(){
    int i, j, temp;
    for(i=0; i<4; i++){
        for(j=0; j<5-i-1; j++){
            if (A[j]>A[j+1]){
                temp=A[j];
                A[j]=A[j+1];
                A[j+1]=temp;
        }   }
    }
    printf("SORTED ARRAY");
    prarf("%d", A[i]);
}
void divisible(int m){
      int i=0;
```

```
    while (i<5) {
        if (A[i]%m==0){
            printf("%d", A[i]);
        }
        i++;
    }
}
```

5) Binary search using recursion.

```
#include <stdio.h>
#include <stdlib.h>
void Binarysearch (int arr[], int num, int fir,
                   int las);
    if (fir>las){
        printf ("In not found in array");
    }
    else{
        int mid;
        mid = (fir+las)/2;
        if (arr[mid]==num){
            printf ("Elemend is found at %d",mid+1)
            exit(0);
        }
        else if (arr[mid]>num){
            BoharySearch(int arr[], inl num,
                         fir, mid-1);
```

```
    }
    else {
        BinarySearch(int arr[], int num, mid+1,
                     int las);
    }
void main(){
    int arr[] = {2,5,7,9,12,15,18}.
    int num = 9.
    int first=0, last = (sizeof(arr)/sizeof(arr[0]));
    BinarySearch (arr, num, first, last);
}
```