C - Assignment - 6

Nandini Chalicham
CSE - F
-AP19110010333

1, (a) (b)

```
#include<stdio.h>
int binary search,(int arr[ ], int a, int b, int x);
{
  if (b>=a)
  {
   int mid = a + (b-a)/2;
   if (arr[mid] ==x)
   {
    return x;
   }
   if(arr [mid ]>x)
    return binary seach(arr, a, mid-1, x);
  return binary search(arr, mid+1, b, x);
  }
   return -1;
  }
    int main()
  {
   int num;
   printf("Enter size of array: ");
   scanf ("%d", &num);
   int i, j, a, val[num], val1, s, s2, sum;
   for(a=0; a<num; a++)
   {
    printf (" Enter value: ");
    scanf("%d", & var[a]);
   }
  for (i=0; i<num; i++)
  {
```

```
    for (j = i+1; j<num; ++j)
    {
      if (var[i] < var[j])
      {
        a = var[i];
        var[i] = var[j];
        var[j] = a;
      }
    }
  }
  printf("Array in DO: ");
  for (i=0; i<num; i++)
  {
    printf("%d", var[i]);
  }
  printf("\n List \n");
  printf("1. Find value at entered position \n 2. Find position
          of element \n 3. Print sum & multiply 2 values");
  printf("\n Enter choice: \n");
  scanf("%d", &OP);
  switch(OP)
  {
  case 1:-
  printf("Enter position: ");
  Scanf("%d", &var);
  printf("The value of %d position is %d", var, var[var]);
  break;
  case 2:
  printf("Enter element to find position:");
  Scanf("%d", var);
```

```
int result = binary
(result == -1)
{
printf ("element is not present in array");
}
return 0;
case 3:
printf ("\n Enter two positions to find sum & product of values \n");
scanf ("%d %d", &s1, &s2);
sum = var[s1] + var[s2];
pro = var[s1] + var[s2];
printf ("sum = %d \n", sum);
printf ("Multiplication = %d", pro);
break;
}
}
```

(2)

```
#include <stdio.h>
#include <stdlib.h>
void merge(int arr[], int a, int b, int c)
{
int x, y, z;
int n1 = b-a+1
int n2 = c-b
int A[n1], C[n2];
for (i=0; i<n1; i++){
A[i] = arr[a+i];
}
for (j=0, a<z; j++){
C[j] = arr[b+1+b];
}
a=0;
```

```
while (i<n, && j<n2) {
 if (A[i] <= C[j]) {
   arr [k] = A[i];
   a++;
 }
 else{
  arr [b] = c[b];
  b++;
 }
  k++;
}
void merge sort (int arr [], int a, int c)
{
  if (a<c)
  {
   int b = a+(c-a)/2;
   merge soot (arr a, b);
   merge sort (arr, b+1, c);
   merge (arr, a, b, c);
  }
}
  void print array (int B[]; int size)
  {
   int a,
   for (a=0; a<size; a++)
    {
     int a;
     for (a=0; a<size; a++)
     {
      printf ("%d", B[a]);
      printf ("\n");
     }
```

```
printf ("Given array is \n");
printArray(var, size);
mergeSort(var, 0, siz - 1);
printf("\n Sorted array is \n");
printArray(var, size);
int k, f, a, P1, P2 temp,
printf("Enter the value of k to the product of elements
        from first & last : ");
scanf ("%d", &k);
P1 = P2 = 1;
for(f=0; f <= k; f++){
temp = var[f];
  P1 += temp;
}
for(a=size-1; a >= k; a--)
{
 temp = var[a];
 P2 += temp;
 printf(" product of elements : %d %d", P1, P2);
```

}

(3) Selection sort:-

It performs sorting by searching for the min value number and placing it into the first or last position according to the order. This process of searching the min key and placing it in the proper position is contd. until all elements are placed at right position.

Ex of ~~select~~ Insertion sort:-

4 3 2 10 12 1 5 6

3 4 2 10 12 1 5 6

2 3 4 10 12 1 5 6

1 2 3 4 10 12 05 6

1 2 3 4 5 10 12 6

1 2 3 4 5 6 10 12

Ex of selection sort:-

		min value at index
i = 0	20 12 10 15 2	1
i = 1	20 12 10 15 2	2
i = 2	20 12 10 15 2	3
i = 3	20 12 10 15 2	4

2 10 12 15 20

swapping

Insertion sort:-

It works by inserting the set of values in the existing sorted side. It constructed the sorted away of inserting a single element at a time. This process continues until whole array is sorted in same order. The primary concept behind insertion sort is to insert each item into its appropriate place in the final list.

(14)
```
#include <stdio.h>
void bubble sort (int arr[ ], int n)
{
 int i , j, temp;
 for (i = 0; i<n, i++)
 for(j =0; j<n-i-1;j++ ){
 if [arr[j] > arr[j+1])
 {
   temp = ar[j];
   ar[j] = ar[j+1];
   ar[j+1] = temp;
  }
}
int main(){
int size i;
printf (" Enter the size: \n) ;
scanf (" %d ", &size);
int arr [size];
for (i=0; i < size; i++)
{
printf ("Enter element : ");
scanf ("%d", & arr[i]);
}
bubble sort (arr, size);
printf (" sorted array: \n");
for (i=0 ; i < siz; i ++)
{
  printf(" %d", arr [i]);
   printf ( " \t");
}
```

```
printf("\n Menu\n");
printf(" 1. Display elements in alternate order: ");
printf(" 2. Sum of elements in odd positions and product
          of elements in even positions\n");
printf(" 3. Divisible by m\n");
int op, sum=0, product=1, m;
printf(" Enter choice: ");
scanf("%d", &op);
switch(op){
case 1:
 for(i=0; i<size; i+=2){
  printf("%d\t", arr[i]);
 }
 case 2:
 for(i=0; i<siz; i+=2){
  sum = sum + arr[i];
 }
  for(i=1; i<size; i+=2){
  product = product + arr[i];
  }
  printf("sum: %d\n", sum);
  printf("product: %d\n", product);
 case 3:
  printf(" Enter m value: ");
  scanf("%d", &m);
  printf(" No's divisible by %d are: \n", m);
  for(i=0; i<size; i++){
   if(arr[i] % m == 0){
    printf("%d\t", arr[i]);
```

⑤
```
#include<stdio.h>
int binarysearch(int l, int h, int x)
{
 int mid=(l+h)/2;
 if(l>h) return -1;
 if(a[mid]==x)
 return mid;
if(a[mid]<x){
return binarysearch(a, mid+1, h, x);}
 else{
 return binarysearch(a, l, mid-1, x);
}
int main(void)
{
 int a[100];
 int siz, pos, val;
printf("Enter the length of array: ");
scanf("%d", &siz);
printf("\n Enter array elements \n");
for(int i=0; i<siz; i++){
scanf("%d", &a[i]);
printf(" Enter element to search\n");
scanf("%d", &valu);
pos= binarysearch(a,0, size-1, val);
if(pos<0){
  printf("cannot find the element %d in array\n", val);
 }
 else{
 printf("The position of %d in array is %d\n", val, pos+1);
 return 0;
}
```