Y<sub>i</sub>Nikhil APMII∞losi4 - CSE - G

) Take the elements from the user and sort them in decending earder and do the following.

a) Using binaxy seasch find the element and the location in the assay where the element is asked from users in the user to entex any two locations point the sum and product of values at those locations in

the sooted assay.

# include a stations

int main()

int 1, bu, high, mid, n, key, axx[100], temp, 1, one, two,

paints ("Enter the number of elements in assay");

scanf (" > d", 4n);

prints ("Enter > d integers," n);

prints ("Enter > d integers," n);

scanf (" > d", 4 arr [i]);

scanf (" > d", 4 arr [i]);

```
( apid + worl) = bin
   for (1=0; icn; i++)
                                    (Apt an) stay
       if (j=i+1; j<n; j++)
          if (0000 [i] < 0000[i])
    if (temp = exx[j]);
                        ( goa = [ bim] oza ) 4 och
             ( [i] = 0xx [i] ;
   cos(j) = temp; to bout to the
                                          100d
        3
paintf ("In elements of assory is sosted in decreasing
                             20000 1(n) )
& (i=0; i<n; i++)
    printf (" 7.d", ax[i]);
  printer ( ) thereon there is bought too, ") though
 1
 paintf ("Enter value to find");
 Stant ("">d", & key);
the = other man in the state out to the " ) the seq
  high = n-19 to
```

```
(++1 '051 0-1)
 mid = (low + high)/; ;
while (bw < high)
                          ( + 10 1 / 1 + ( + ( ) ) ) ( ) ( ) ( ) ( )
E
    if (asx[mid] > key)
                          ([1] 500 > [1] 500) 91 S
     low = mid + 1
                          ((liter may) );
      else if (axo [mid] = key)
                     : [i] can = [j] rac
      paints (" " d sound at location r.d", tey, mid+1);
   bocak;
z
   else
   high = mid - 1; testor is proved it
    aid = (but high)/2;
  if (bus = high)
  { i([] one (b) ") thing
   pointed ("Not found; Y.d isn't possent in the list. n",
                        part ("Inter value to 1913");
   printf (" /n");
                                   (604 4 12 Fred) Fred
 points ("Enter two locations to find sum and product
                                of the elements ")
                                      Scanned with CamScanner
```

```
scanf (" 1.d", fone);
    scanf (" 1/d", ftwo);
                                                1
                        ston of elements - 67
    sum = (axx [one] + axx [two]);
                           attimate to believe
     product = ( our [one] x our [two]);
 paintf (" The sum of elements = 1/d", sum);
printf("The product of elements = 1/d", product);
  setuan o; and els a seria deal ino design
                               < A sibile a shortoni 14
 output =
                                S Harris Showing
      number of elements in array x 5
  Entex 5 integes 2 ([ass xan] too again bar
            (bri toi toi toi) posso open biou
  7
                         the sand from som that
  5
  4
  2
 Element of away is sorted in descarding order.
 97542 Enter value to find 5.
  5 found at location 3 miles by south of
                                     Scanned with CamScanner
```

```
( (Sas p. "+ " ) France
      2
      4
                                      sconf (" 1d", (two);
                of elements = 67
     The
                              ( ( ( ) are fore) + cost pro)) = are
         product of elements = 10
                             seduct = ( our [ope] x care[ture]);
            the away using merge sort, where elements
       Sost
  2)
             from the product of the sunth elements from
            and last where k is taken from the usex.
  A) # include < stalio h >
        # indude conio.h >
                                                     tugter
       # define MAX LISTE 501 themself to setting
       ( [aste - AAM] troe - spen bow
                                         bites s integras
       void mage - accord (int, int, int, int);
      int one - sort [HAX_SIZE];
      int main()
        int 1, E, pro = 1;
       parate (" sample mege sort example functions and assay ")
      pointf (" In Enter 1'd Elements for souting In"
bronch to blood, me him at continue MAX_STZE) in
```

```
for (1=0; 1< MAX_SIZE; 1++)
                           (( Int I Int) took of my blow
E
  Scarf (" 1, d", + our _ sort [1]);
   paint f (" on your data: ");
   POS (1=0; 1 = MAX_ SIZE; 1++)
                                   : N((+1) = m
    printf (" It 1/d", axx_sout [1]); (m; ) tor your
                             (( 100) tro agreen
  3
 merge _ sort (0, MAX_SIZE-1);
 printf (" In sorted data: ");
                          (( mo m 1) force from
 fox (1=0; 1 < MAX_SIZE; 1++)
  parte (" 14 x d" are sort (1);
3
paintf("find the product of the kith element from
                           last where t In");
scort (" 1,4", (+);
                                (b s b) doil stide
Bo = 0xx _ sort [K] + 0xx - sort [MAX _ sIZE-E-];
Mante (" Produce = 1.d", Pap);
```

```
Contraction of a
   void merge_sout (inti, int))
                                                                                                                                Miles me by
£
              int m;
              if (i < i) fi
                                                                                                Constitution of the state of th
                             m = (1+))/2;
                             meage _ sout (i,m); (i)
                           merge_sort (m+1, i);
          11 wending two askeds.
                             merge _ axisay (1, m, m+1, 1);
           3
       void merge assay (inta, intb, intc, int )
                            int 1 = a, 1 = c, K = 0
      while (ich fd i = d)
                                               The same statement of the same of
                      if (cur _ sort [i] < out _ sort [i]) }
                                                                                              000 - 500+ [i++];
```

```
improvedsentis the notate the few notation sound (c
        + [x++] = ar - sort [++];
 LEADER DO THE SAME PROCESSION OF THE PORTURE
 11 collect memoining elements
white while of a mond of the state of the state of
+[k++] = 0xx - 20x+[j++];
   for (i=a, j=a, ic=d; i++1, j++) notein but
are _sost[i] = t[i]; both off or socie storming of
        method over on effective namount of memory
   3
                          that notificant to prichable
 Emple merge sost example - functions and assay
  Enter 5 elements for Sorting is south tour and between
  strong and gradt fitting educat meltinoph guitan all c-
7
                           the between of in
 14
      alternate there are 'n' numbers elements
 6
  2
                                   manched Let
 your data? 7 7 4 6 216 printed west amusic stale
 sorted data: 24679 til all to mind
 the product of 1th elements from this and lest
```

ALC: N

3) Discuss inscrition sort and selection sost with examples

1[1++] = 100 = (++1)+

Insertion Sort :

Insertion sort works by inserting the set of values in the existing sosted file-It consists the socked laxay by inserting a single denen't at a time. This i process it control until whole array is sorted in same order 1 The primary encept behind insertion sout is to insert each item in its appreciate place in the final list. The inscrition sost method somes an effective amount of memory.

## Working of Insertion Sost;

- It uses two sets of arrays where one stores soxted data and other on unsoxted data. ted again styl
- -> The sorting algorithm wasts until there are in the unsorted set.
- -> Lets assume these are 'n' numbers elements in the unsorted set.
- -> lets assume these sensining elements wase in the consoli at the first state of the first state of the position of the
- The first element of the unsorted position has away index 1 (if LB=0) tomes

  Scanned with CamScanner

- After each iteration, it chooses the first determent the insorted position and inserts it into the place in the sorted set. Advantages of Insertion Sort:

- facily implemented and very efficient when wed with and an in terms and and bar a

deta. of sets 11pmz

-> The additional memory space requirement of inscrition 03 11 T3 9h is less (i.e, (o(1))).

is ansidered to be line scoting techniques has the - 7t the new elements use seccived. list can be sorted as

- It is faster than other sorting algorithms. mondain set both n

Complexity of Inscrition Sort & 1 m principal

best case complexity of insertion sort is, o(n) times i.e, when the away is previously sorted. In the some way, when the assay is souted in the severce order the first element in the unsorted opposit sorted set. be composed with each element -fre in 1.) insention of so, in the worst case, running time

is ignadratic in i.e, of (o(n2)) to Inti average case in also it to make the minimum h(k-1)/2 , comparisons. Hence the average case also has quadratic sunning time of titud national to exportage Example = 11 20 196 22 implemented and very efficient Il find the minimum element in are [0--- ord place and sets of data. trat beginning. to transcripe sure positive of of the minimum element in are [1. -- 4] and place place of the minimum element in are [1. -- 4] and place Harriet segments of oxo [1----4] see batroos as non the 11 Find the minimum element in ass [22---4] of and place at beginning of asse [2---4] too? without to private The test cope anytherity of 25 inductions wif is said Il find but the uninimon, element was in one [3. - . v] and place at the beginning of the away [2... y] of a person between sell of towner their sell sell the batter of the state that the begins of took could be south given some of insettion south

Selection Sout ? 1911 have no the same from

The selection wast perform sorting by seasching few the minimum value number and placing it into the flust as last position according to the order (ascending or descending).

The process of seasoling the minimum key and placing it in the proper position is continued until all the elements are placed at sight position.

Working of the selection sort:

1 the selecti

- is suppose an assay with a elements in the memory.
- In the first pass, the smallest key is searched along with it's position, then the Arr [ros] is supposed and succepted, with Arr [D]. Therefore Arr[D] is sorted.
- usive is determined in the sub oursay of (n-1) elements inter change the non-[pos] with Abr[1].
- In the pass (n-i), the same process is performed to

Advantages of selection soist:

The main advantage of sulection sout is that is it

performs well on a small list. - Further more, because it is to an in place scoting, algor No additional temporary testosage dans is alsequired a beyond what is needed to hald at the original militist. the possess of sensething the minimum key and pasing it of selection sout: As the working of selection sost does not depend on the oxiginal oxder of the elements in the assay s there is not much difference between test case point in ase complexity of selection sort. The selection sort selects gnote behower ei the minimum value element, in of the 22 selection of process. As # pare possibles 'n' number of elements are scanned, these fore in (n-1) compasisions are made in the flost pass. Then, the elements are inter changed. Similarily in the second pass also to find the second smaller element we secquise scanning of rest n-elements and the process is entinued till the whole array sorted. Thus running time complexity of selection sout is  $O(n^2) = (n-1) + (n-2) + - - + 2 + 1$  $\frac{1}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} + \frac{\rho(n-1)}{2} \cdot \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} + \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} + \frac{\rho(n-1)}{2} = \frac{1}{2} \frac{\rho(n-1)}{2} + \frac{\rho(n-1)}{2} = \frac{\rho(n-1)}{$ 

int was [27] . ), in temp, am = 0 pacturet = 1; Example: paintf(" fiter total mates of terp its to 12 13 loop for i=1 (second element of the assay) to 4 (last element of the Cossay) strange by solid " I thing i=1, since 12 is smaller than 13, move 18 and insert enct for to a fox (17); 12 before ١3 ، some for 1=2, 1=3, 1=4 pries porter primes of " Horizon Sosted axeay 6 7 12 13 - 14 (-1) Sext the assay using bubble sost where elements assertation and display the elements. vzex. the from in alternate oxded (i) (1ii) sum of elements in odd positions and products of in even positions. elements Potilisan - Pilaa which are divisible by m where m is taken (iii) elements from the user. # include < stdio.h > A) # include canio.h > int man() fill of my who prilitizes of strong war " hand

```
int our [sa], i, j, n, temp, som =0, product =1;
                  printf ("Entex total number of elements to store:")
         of scanf (" V.d", 4h); thenest tones? ) 1 1 of pois
                  print ? (" Enter V.d elements (" ") ill to tombe tout)
        fox (1=0, i&n; itat) (1) not reller in sold
              sout (".19", focs [1]);
                  print? (" In sorting array using bubble sort technique in");
             € (i=0', i < (n-1)', H++); 31 1 3
                                                                                                                                          more kotes .
for (1=0; ) < (1-1) 1, i++) id
           { if one (1) > one (1+1) the fait pur sace of
 temp = axx [i] it is it is a strange to me or
                                                                                                                      edes showsta of (i)
                                        coeff = coef
                                            ass[it] = temp; sidesoit are thirty extensive (1)
restrict a m
              3
                                                                                                                                     Tri white should it
               3
              point f (" All manay elements sorted successfully in");
              prints ("Array elements in ascording order: In In);
```

```
flox (i=0; icn; i++)
                                             there
     wintf (" " dln", and [1]);
                                  2th ab 2 yetal
point of "assay elements in attemate order (n");
  for (i=0; i==n; i=1+e)
     point ( " v.d" /n", ass [1]);
    3
  for (i=1; ic=n 1 = i+2); where strange proc
               way dements in exercing rober
    {
     Sum = sum + acre [i];
    3
 panter (" the sum of old position dements are = 1/d /n", sum);
  for (1=0; i = 1, i=1+2)
                  devents in thought today
      product = aso [i];
   3
  point of the products of even position elements are 1 d in product)
  get in (); we transme such as the super) soil
  setum o ();
```

```
105 (1=0 | 100 | 1++)
  output?
 Enter total number of elements to store = 5.
                         printf (" 1 d (n ) 00 [1]);
  Enter 5 clements
           part (" assert demonts in attende order In");
                               (i-0) i = 1+2)
   Ч
    3
                       wintf (" v.d. vo") ass [1]);
    2
   Sosting array using bubble sost technique.
        away elements sorted successfully
   hosay delements in ascending order
                               sum = sum + and il
    2
Mitted (" the sum of old position depents are = /a /o p vem)"
   array elements in atternate
      2
                                 potest = and [13]
      4
      ଟ
had The sum dofiels addica position elementations of ") that
    The product of over position dement are 16,4.16
```

```
implement
            remazine brodrow
                              ь
                                            binaxy
   Waite
                                                   sassify 3
5)
          ۵
     # include < staio.h>
A)
     # include < conio (4301, 1 Lim tour 180) downe provid
        binosy seasch (int ass[], int num, int first, int last)
   E
     int mid;
      if (fast > last)
                                      ( ) sim bini
        printf ("Number is not found");
      ٤
             mint? ("Lotes The size of an array")
     else
    if (ar [mid] == num)
                             (++1 : 0>1 (0=1) sof
    point P(" element is found - of index /d", mid);
                                               10
    exit(0);
  3
   else if (arr[mid] > nom)
   primary search (arr, num, first mid-);
```

```
from the a recordine proposer to implement binary
  else
                                       SH oilete & shuishi |
   ٤
            search (axx; num, mid +1, last); in = shuton; !
 (beat this test this num that I see that ) there people Bloo
    3
 3
3
                                      (teal - teat) H
    void main()
                   ("bound for a some ") I think
   int axx [100], bog, mid, end, i, n, num;
  print P ("Enter the Size of an array");
                                                  9319
  Scanf (" 1.d", 4n);
   point f ("Enter the value in sorted sequence (n");
 物× (i=o; ien; i++)
                                 (mun - [him] ma)
     scanf (" 1.d", foxs [i]); bound & formats ] + then
   13
   beg = 0
                               (mon - [bim/mod] & sels
   end = n-1;
  point of "Enter the size of an assay");
                                            Scanned with Camscanner
```

```
scarf (" 1.d", 4n);
  paints ("Enter the values in souted sequence In");
for (i=0; i=n; i++)
    sunf (" 1/4", 4 0xx [i]);
  3
 bg = 0
  end = n-1;
  print & (" Enter a value to be seasch: ");
  sant (" 1.d", enm);
 Binaxy seasch (axe, num, beg, end);
Output:
Enter the size of an array 5
Enter the value in sorted sequence
 4
 5
 7
       a value to seasch: 5
 Element is found at index :1
```