1] Find and return the maximum and minimum of n given numbers.

#include<iostream>

#include<conio.h>

using namespace std;

int max = 0;

int min = 0;

void maxmin(int i, int j,int arr[])

{

int mid, max1, min1;

if (i == j)

max = min = arr[i];

else if (i == j - 1)

{

if (arr[i] < arr[j])

{

max = arr[j];

min = arr[i];

}

else

{

max = arr[i];

min = arr[j];

}

}

else

{

mid = (i + j) / 2;

maxmin(i, mid, arr);

max1 = max;

min1 = min;

maxmin(mid + 1, j, arr);

if (max < max1)

max = max1;

if (min > min1)

min = min1;

}

}

void main()

{

int a, large, least, num, arr[100];

cout << "Enter the number of elements " << endl;

cin>>num;

cout << "Enter elements " << endl;

for (a = 0;a < num;a++)

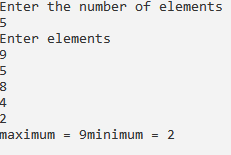
cin >> arr[a];

maxmin(0, num - 1, arr);

cout << "maximum = " << max << "minimum = " << min << endl;

\_getch();

}



2] Finding the kth smallest element.

#include<iostream>

#include<conio.h>

using namespace std;

// Swapping two values.

void swap(int \*a, int \*b)

{

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}

// Partitioning the array on the basis of values at high as pivot value.

int CreatePartition(int a[], int low, int high)

{

int pivot, index, i;

index = low;

pivot = high;

// Getting index of pivot.

for (i = low; i < high; i++)

{

if (a[i] < a[pivot])

{

swap(&a[i], &a[index]);

index++;

}

}

// Swapping value at high and at the index obtained.

swap(&a[pivot], &a[index]);

return index;

}

// Implementing Partition.

int Partition(int a[], int low, int high, int k)

{

int pindex;

if (low < high)

{

// Partitioning array using last element as a pivot.

// Recursively implementing partitioning in the direction to place the pivot at (k-1)th pivot.

pindex = CreatePartition(a, low, high);

if (pindex == k - 1)

return k - 1;

else if (pindex > k - 1)

Partition(a, low, pindex - 1, k);

else

Partition(a, pindex + 1, high, k);

}

}

int main()

{

int n, i, k, kk;

cout << "Enter the number of element: " << endl;

cin >> n;

int arr[100];

cout << "Enter element in array " << endl;

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

cin >> arr[i];

cout << "Enter the k for the kth smallest element: " << endl;

cin >> k;

kk = Partition(arr, 0, n - 1, k);

// Printing the result.

cout << "The kth smallest element: " << arr[kk] << endl;

\_getch();

return 0;

}

3] Generate the Fibonacci series recursively.

#include<iostream>

#include<conio.h>

using namespace std;

void printFibonacci(int n)

{

static int n1 = 0, n2 = 1, n3;

if (n>0)

{

n3 = n1 + n2;

n1 = n2;

n2 = n3;

cout << n3 << " ";

printFibonacci(n - 1);

}

}

int main()

{

int n;

cout << "Enter the number of elements: ";

cin >> n;

cout << "Fibonacci Series: ";

if(n==1)

cout << "0 " << endl;

else if(n==2)

cout << "0 1 " << endl;

else

{

cout << "0 1 ";

printFibonacci(n - 2);

}

\_getch();

return 0;

}

C:\20CE14\20CE14 MADF LAB\10\3.png

4] Find the nth number in the Fibonacci series recursively.

#include<iostream>

#include<conio.h>

using namespace std;

int fib(int n)

{

if (n == 1)

return 0;

else if (n == 2)

{

return 1;

}

else

return fib(n - 1) + fib(n - 2);

}

int main()

{

int n;

cout << "Enter element number" << endl;

cin >> n;

cout << "" << n << " element is: " << endl;

cout << fib(n);

\_getch();

return 0;

}

