Array Problems.

1. Odd and Even segregation

2 1 8 4 3 6 9 5

2 8 4 6 1 3 9 5

Solution:

#include <stdio.h>

int main()

{

int SIZE, ind, itr;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

int limit = SIZE;

ind = 0;

while(ind < limit)

{

if( arr[ind] % 2 == 1 )

{

int temp = arr[ind];

for(itr = ind; itr < SIZE - 1; itr++)

arr[itr] = arr[itr + 1];

arr[SIZE - 1] = temp;

limit--;

}

else

ind++;

}

for(ind = 0; ind < SIZE; ind++)

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

return 0;

}

1. Selection Sorting

5

3 5 2 1 4

1 2 3 4 5

Solution:

#include <stdio.h>

int main()

{

int SIZE, ind, ind1, ind2;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

for(ind1 = 0; ind1 < SIZE - 1; ind1++)

{

int min\_ind = ind1;

for(ind2 = ind1 + 1; ind2 < SIZE; ind2++)

{

if( arr[ind2] < arr[min\_ind] )

min\_ind = ind2;

}

if(min\_ind != ind1)

{

int temp = arr[min\_ind];

arr[min\_ind] = arr[ind1];

arr[ind1] = temp;

}

}

for(ind = 0; ind < SIZE; ind++)

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

return 0;

}

1. Insertion Sort

5

5 4 3 2 1

1 2 3 4 5

\*/

#include <stdio.h>

int main()

{

int SIZE, ind, ind1, ind2, itr;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

for(ind1 = 0; ind1 < SIZE - 1; ind1++)

{

int min\_ind = ind1;

for(ind2 = ind1 + 1; ind2 < SIZE; ind2++)

if( arr[ind2] < arr[min\_ind] )

min\_ind = ind2;

if( min\_ind != ind1 )

{

int temp = arr[min\_ind]; // pick

for(itr = min\_ind ; itr > ind1 ; itr--) // right shift

arr[itr] = arr[itr - 1];

arr[ind1] = temp; // insert

}

}

for(ind = 0; ind < SIZE; ind++)

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

return 0;

}

1. Frequency of digits in a number.

122333445

1 -> 1

2 -> 2

3 -> 3

4 -> 2

5 -> 1

Solution

#include <stdio.h>

int main()

{

int freq[10] = {0};

unsigned long long int num;

scanf("%llu", &num);

while( num )

{

int dig = num % 10;

freq[ dig ]++;

num /= 10;

}

for( int ind = 0; ind < 10; ind++)

if( freq[ind] != 0)

printf("%d -> %d\n", ind, freq[ind] );

return 0;

}

1. Second minimum element in a given array. ( without sorting the array).

7

1 2 3 4 5 6 7 23 6 31 64 32 53 77

2 23

\*/

#include <stdio.h>

int main()

{

int SIZE = 10;

int arr[] = {54,12,78,23,65,1,66,13,76,44};

int min, sec\_min, ind;

if( arr[0] < arr[1] )

{

min = arr[0];

sec\_min = arr[1];

}

else

{

min = arr[1];

sec\_min = arr[0];

}

for(ind = 2; ind < SIZE; ind++)

{

if( arr[ind] < min )

{

sec\_min = min;

min = arr[ind];

}

else if( arr[ind] < sec\_min )

sec\_min = arr[ind];

}

printf("%d", sec\_min);

return 0;

}

6.

/\*

Find whether the given array is Monotonic or Not.

1 2 6 8 21 43 56 77 88 91 -> Monotonic

2 4 8 21 45 23 12 67 87 88 -> Not Monotonic

98 76 54 44 32 28 25 21 12 8 -> Monotonic

\*/

Solution:

#include<stdio.h>

int main()

{

int SIZE, ind, just, temp;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

if( arr[0] < arr[1])

{

for(ind = 1; ind < SIZE - 1; ind++)

{

if( arr[ind] > arr[ind+1])

{

printf("Not Monotonic");

return 0; // end of program.

}

}

}

else

{

for(ind = 1; ind < SIZE - 1; ind++)

{

if( arr[ind] < arr[ind+1])

{

printf("Not Monotonic");

return 0;

}

}

}

printf("Monotonic");

return 0;

}

7.

/\*

Multiply previous and next element

No previous - 0

No next - 1

6

1 2 3 4 5 6

0 3 8 15 24 5

\*/

Solution:

#include<stdio.h>

int main()

{

int SIZE, ind, just, temp;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

for(ind = 0; ind < SIZE; ind++)

{

if( ind == 0 )

{

temp = arr[ind];

arr[ind] = 0;

}

else if( ind == SIZE - 1)

{

arr[ind] = temp;

}

else

{

just = arr[ind];

arr[ind] = temp \* arr[ind+1];

temp = just;

}

}

for(ind = 0; ind < SIZE; ind++)

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

return 0;

}