Warm – Up 1)

#include<stdio.h>

void swap(int \*ptr1, int \*ptr2);

int main()

{

int a = 10;

int b = 2;

int \*ptra=NULL;

int \*ptrb=NULL;

ptra = &a;

ptrb = &b;

printf("Values of a and be BEFORE SWAPPING\n");

printf("a = %d and b = %d\n", a,b);

swap(ptra,ptrb);

printf("\nValues of a and be AFTER SWAPPING\n");

printf("a = %d and b = %d\n\n", a,b);

return 0;

}

void swap(int \*ptr1, int \*ptr2)

{

int temp;

temp = \*ptr1;

\*ptr1 = \*ptr2;

\*ptr2 = temp;

}

Warm Up 2)

#include<stdio.h>

void sort3Int(int \*num1, int \*num2, int \*num3);

int main()

{

int a = 2;

int b = 3;

int c = 1;

int \*ptra=&a;

int \*ptrb=&b;

int \*ptrc=&c;

printf("Values of a,b and c BEFORE SORTING\n");

printf("a = %d b = %d c = %d\n", a,b,c);

sort3Int(ptra,ptrb,ptrc);

printf("\nValues of a,b and c AFTER SORTING\n");

printf("a = %d b = %d c = %d\n", a,b,c);

return 0;

}

void sort3Int(int \*num1, int \*num2, int \*num3)

{

int temp;

//1st case num1 smallest

if(\*num1<=\*num2 && \*num1<=\*num3)

{

if(\*num2<=\*num3)

return; // num1<num2<num3

else// num1<num3<num2

{

temp = \*num3;

\*num3 = \*num2;

\*num2 = temp;

}

}

//2nd case num2 smallest

else if(\*num2<=\*num1 && \*num2<=\*num3)

{

if(\*num1<=\*num3) // num2 < num1 <num3

{

temp = \*num1;

\*num1 = \*num2;

\*num2=temp;

}

else // num2< num3 < num1

{

// num2 < num1 <num3

temp = \*num1;

\*num1 = \*num2;

\*num2=temp;

//num2< num3<num1 (swapping num3 and num1)

temp = \*num3;

\*num3 = \*num2;

\*num2 = temp;

}

}

else // 3rd case num3 is the smallest

{

if(\*num1<=\*num2) // num3< num1 < num2

{

//num3<num2<num1

temp = \*num1;

\*num1 = \*num3;

\*num3 = temp;

//num3< num1<num2

// because here num3 holds the bigger number

temp = \*num3;

\*num3 = \*num2;

\*num2 = temp;

}

else // num3 < num2 < num1

{

temp = \*num3;

\*num3 = \*num1;

\*num1 = temp;

}

}

}

Warm Up3)

#include<stdio.h>

#include<limits.h>

void range(int \*x, int size, int \*maxPtr, int \*minPtr);

int main()

{

int a[] = {5, 8, 23, 12, 7, 16, 19};

int size, i, min, max;

size = sizeof(a)/4;

printf("Array a has %d elements\n", size);

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

printf("%d ", \*(a+i));

printf("\n\n");

range(a,size,&max,&min);

printf("\nThe max value of the array %d\n", max);

printf("\nThe min value of the array %d\n", min);

return 0;

}

void range(int \*x, int size, int \*maxPtr, int \*minPtr)

{

int i;

\*maxPtr = INT\_MIN;

\*minPtr = INT\_MAX;

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

{

//Finds the max

if(\*(x+i) > \*maxPtr)

\*maxPtr = \*(x+i);

//Finds the min

if(\*(x+i)< \*minPtr)

\*minPtr = \*(x+i);

}

}

Warm Up4

#include<stdio.h>

#include<limits.h>

#define SIZE 55

void outputArray(int \*arr,int element, int length, int k);

int main()

{

int values[SIZE];

int size, index=0, num, len, k;

printf("Please enter int to fill array. To stop enter negative number\n");

while(1)

{

scanf("%d",&num);

if(num<0)

break;

\*(values+index) = num; // same as values[index]=num;

index++;

}

printf("Enter the # of elements to print per line [1 10]\n");

scanf("%d",&len);

// validation liip

while(len<1 || len>10)

{

printf("%d is out of the interval. Enter [1 10]\n",len);

scanf("%d",&len);

}

printf("Enter the index number you want to print from\n");

scanf("%d",&k);

printf("Values of Array\n\n");

outputArray(values,index,len, k);

printf("\n\n");

return 0;

}

void outputArray(int \*arr, int element,int length, int k)

{

int i;

for(i=k-1;i<element;i++)

{

printf("%d ", \*(arr+i)); //arr[i]

if((i+1)%length==0)

printf("\n");

}

}