Question 1

CODE

/\*The values of j ank k are not swapped because of wrong swapping

it should be done like int tempt=a then a=b and then b=temp\*/

//Also swapping does not occur because the changes are made only in the local variables thus we have to use pointers

#include<stdio.h>

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

int temp=\*a;

\*a=\*b;

\*b=temp;

}

void main(void){

int j=2, k=5;

int \*pt1=&j, \*pt2=&k;

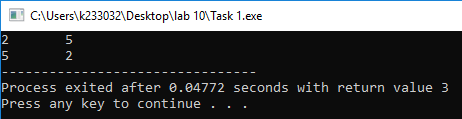
printf("%d\t%d\n",j,k);

swap(pt1,pt2);

printf("%d\t%d",j,k);

}

OUTPUT



Question 2

CODE

#include<stdio.h>

void reverse(int \*array, int size);

void main(void){

int i, size;

printf("Enter the size of the array:\n");

scanf("%d",&size);

int array[size];

printf("Now enter the elements of the array:\n");

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

scanf("%d",&array[i]);

}

int \*ptr=array;

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

printf("%d ",array[i]);

}

printf("\n \n");

reverse(array,size);

}

void reverse(int \*array, int size){

int i;

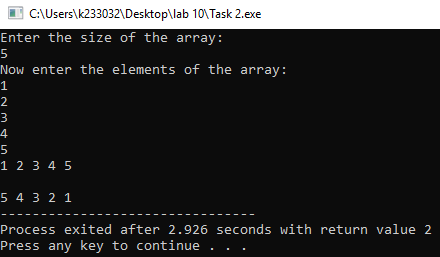
for(i=size-1; i>=0; i--){

printf("%d ",array[i]);

}

}

OUTPUT



Question 3

CODE

#include<stdio.h>

void main (void){

//adding to pointers of different types works differently because \*ptr=iterator+(sizeofdatatype)

int N1=0,N2=0,N3=0,i=0;

printf("Enter the size:\n");

scanf("%d",&N1);

char array1[N1];

printf("Enter the values of array of character type:\n");

for(i=0; i<N1; i++){

scanf(" %c",&array1[i]);

}

printf("Enter the size:\n");

scanf("%d",&N2);

int array2[N2];

printf("Enter the values of array of integer type:\n");

for(i=0; i<N2; i++){

scanf("%d",&array2[i]);

}

printf("Enter the size:\n");

scanf("%d",&N3);

long long int array3[N3];

printf("Enter the values of array of long long integer type:\n");

for(i=0; i<N3; i++){

scanf("%lli",&array3[i]);

}

char \*ptr1=array1;

int \*ptr2=array2;

long long int \*ptr3=array3;

for (i=0; i<N1; i++){

printf("%p , %c \n",ptr1,\*ptr1);

ptr1++;

}

printf("\n\n");

for (i=0; i<N2; i++){

printf("%p , %d \n",ptr2,\*ptr2);

ptr2++;

}

printf("\n\n");

for (i=0; i<N3; i++){

printf("%p , %lli \n",ptr3,\*ptr3);

ptr3++;

}

}

OUTPUT

