**22/03/2024**

**Class Tasks:**

1**) Reverse Number:**

**Write a program that prompts the user to enter a positive integer and then prints the number in reverse order**

**Output:**

**without function :**

int main()

{

int number,reverse\_value=0;

printf("enter the number :");

scanf("%d",&number);

while(number!=0)

{

int temp=number%10;

reverse\_value=(reverse\_value\*10)+temp;

number/=10;

}

printf("%d",reverse\_value);

}

**With function :**

#include <stdio.h>

int reverse\_func(int number)

{

int reverse\_value=0;

while(number>0)

{

int temp=number%10;

reverse\_value=(reverse\_value\*10)+temp;

number/=10;

}

return reverse\_value;

}

int main() {

int number;

printf("enter the number :");

scanf("%d",&number);

int return\_reverse\_value=reverse\_func(number);

printf("%d",return\_reverse\_value);

}

**2) Fill in the reverseArray function to reverse the elements of the given array arr in- place.**

**Question:**

#include <stdio.h>

void reverseArray(int arr[], int size)

{

// TODO: Implement array reversal logic here

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

reverseArray(arr, size);

printf("Reversed Array: ");

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

{

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

}

return 0;

}

**Output:**

#include <stdio.h>

void reverseArray(int arr[], int size)

{

int last=size-1;

for(int i=0;i<=size/2;i++)

{

int temp=arr[i];

arr[i]=arr[last];

arr[last]=temp;

last--;

}

}

int main()

{

int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

reverseArray(arr, size);

printf("Reversed Array: ");

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

{

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

}

return 0;

}

**3) Fill in the findMissingNumber function to find the missing number in the given array arr, which contains consecutive integers starting from 1.**

**Question :**

#include <stdio.h>

int findMissingNumber(int arr[], int size) {

// TODO: Implement logic to find the missing number

return -1;

}

int main()

{

int arr[] = {1, 2, 4, 5, 6};

int size = sizeof(arr) / sizeof(arr[0]);

int missingNumber = findMissingNumber(arr, size);

printf("Missing Number: %d\n", missingNumber);

return 0;

}

**Output:**

#include <stdio.h>

int findMissingNumber(int arr[], int size)

{

int i;

for (i = 0; i < size - 1; i++)

{

if (arr[i + 1] - arr[i] > 1)

{

return arr[i] + 1;

}

}

return -1;

}

int main() {

int arr[] = {1, 2, 4, 5, 6};

int size = sizeof(arr) / sizeof(arr[0]);

int missingNumber = findMissingNumber(arr, size);

if (missingNumber != -1)

{

printf("Missing Number: %d\n", missingNumber);

} else

{

printf("No missing number found.\n");

}

return 0;

}

**4) Fill in the rotateArray function to rotate the elements of the given array arr to the right by k positions.**

Question :

#include <stdio.h>

void rotateArray(int arr[], int size, int k) {

// TODO: Implement array rotation logic here

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

int k = 2; // Number of rotations

rotateArray(arr, size, k);

printf("Rotated Array: ");

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

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

}

return 0;

}

**Output :**

#include <stdio.h>

void rotateArray(int arr[], int size, int k)

{

int temp[k];

for (int i = 0; i < k; i++)

{

temp[i] = arr[size - k + i];

}

for (int i = size - 1; i >= k; i--)

{

arr[i] = arr[i - k];

}

for (int i = 0; i < k; i++)

{

arr[i] = temp[i];

}

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int size = sizeof(arr) / sizeof(arr[0]);

int k = 2;

rotateArray(arr, size, k);

printf("Rotated Array: ");

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

{

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

}

return 0;

}

**5) Fill in the findIntersection function to find and print the intersection of two arrays arr1 and arr2.**

**Question :**

#include <stdio.h>

void findIntersection(int arr1[], int size1, int arr2[], int size2) {

// TODO: Implement logic to find and print the intersection of two arrays

}

int main() {

int arr1[] = {1, 2, 3, 4, 5};

int size1 = sizeof(arr1) / sizeof(arr1[0]);

int arr2[] = {3, 4, 5, 6, 7};

int size2 = sizeof(arr2) / sizeof(arr2[0]);

findIntersection(arr1, size1, arr2, size2);

return 0;

}

**Output:**

#include<stdio.h>

void findIntersection(int arr1[], int size1, int arr2[], int size2)

{

for (int i = 0; i < size1; i++)

{

for (int j = 0; j < size2; j++)

{

if (arr1[i] == arr2[j])

{

printf("%d ", arr1[i]);

break;

}

}

}

}

int main(){

int arr1[] = {1, 2, 3, 4, 5};

int size1 = sizeof(arr1) / sizeof(arr1[0]);

int arr2[] = {3, 4, 5, 6, 7};

int size2 = sizeof(arr2) / sizeof(arr2[0]);

findIntersection(arr1, size1, arr2, size2);

return 0;

}

**6) Print the array values**

**Question:**

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4, 5};

printf("%d\n", );

printf("%d\n", );

printf("%d\n", );

printf("%d\n", );

printf("%d\n", );

return 0;

}

**Output:**

#include <stdio.h>

int main() {

int arr[] = {1, 2, 3, 4, 5};

printf("%d\n",arr[0]);

printf("%d\n",arr[1]);

printf("%d\n",arr[2]);

printf("%d\n",arr[3]);

printf("%d\n",arr[4]);

return 0;}