Task#1

Code:

#include<iostream>

using namespace std;

double\* momentum(double velocity[3], const double\* mass) {

double\* ans = new double[3];

for (int i = 0; i < 3; i++) {

\*(ans + i) = velocity[i] \* (\*mass);

}

return ans;

}

int main() {

double velocity[3];

double mass;

cout << "Enter Mass : "; cin >> mass;

for (int i = 0; i < 3; i++) {

cout << "Enter Velocity : "; cin >> velocity[i];

}

double\* ptr = momentum(velocity, &mass);

for (int i = 0; i < 3; i++) {

cout << "Momentum : " << \*(ptr + i) << endl;

}

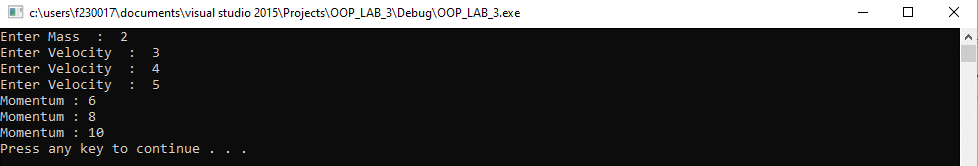
delete[] ptr;

system("pause");

return 0;

}

Output:



Task#2

Code:

#include<iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

double\* randVec() {

srand(time(0));

double\* vel = new double[3];

for (int i = 0; i < 3; i++) {

vel[i] = rand() % 201 - 100;

}

return vel;

}

double\* momentum(double velocity[3], const double\* mass) {

double\* ans = new double[3];

for (int i = 0; i < 3; i++) {

\*(ans + i) = velocity[i] \* (\*mass + i);

}

return ans;

}

int main() {

double\* ans[1000];

srand(time(0));

double sum = 0;

for (int i = 0; i < 1000; i++) {

double mass[3];

for (int k = 0; k < 3; k++) {

mass[k] = rand() % 10 + 1;

}

ans[i] = momentum(randVec(), mass);

for (int j = 0; j < 3; j++) {

sum = sum + ans[i][j];

}

delete[] ans[i];

}

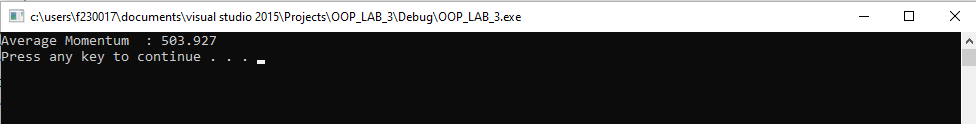
cout << "Average Momentum : " << sum / 1000 << endl;

system("pause");

return 0;

}

Output:



Task#3

Code:

#include<iostream>

using namespace std;

int main() {

int\* arr1 = new int[10];

int\* arr2 = new int[8];

cout << "Array #1" << endl;

for (int i = 0; i < 10; i++) {

cout << "Enter Element : "; cin >> \*(arr1 + i);

}

cout << "Array #2" << endl;

for (int i = 0; i < 8; i++) {

cout << "Enter Element : "; cin >> \*(arr2 + i);

}

cout << "Array #1 : ";

for (int i = 0; i < 10; i++) {

cout << \*(arr1 + i) << " ";

}

cout << endl;

cout << "Array #2 : ";

for (int i = 0; i < 8; i++) {

cout << \*(arr2 + i) << " ";

}

cout << endl;

int arr3[10] = { 0 };

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 8; j++) {

if (\*(arr1 + i) == \*(arr2 + j)) {

\*(arr3 + i) = \*(arr1 + i);

}

}

}

for (int i = 0; i < 10; i++) {

for (int j = i + 1; j < 10; j++) {

if (\*(arr3 + i) == \*(arr3 + j)) {

\*(arr3 + j) = -9999999;

}

}

}

cout << endl;

int flag = 0;

for (int j = 0; j < 10; j++) {

if (\*(arr3 + j) > 0) {

flag++;

}

}

int\* final = new int[flag];

int temp = 0;

for (int j = 0; j < 10; j++) {

if (\*(arr3 + j) > -9999999) {

\*(final + temp) = \*(arr3 + j);

temp = temp + 1;

}

}

cout << "Intersection : ";

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

cout << \*(final + i) << " ";

}

cout << endl;

delete[] arr1;

delete[] arr2;

delete[] arr3;

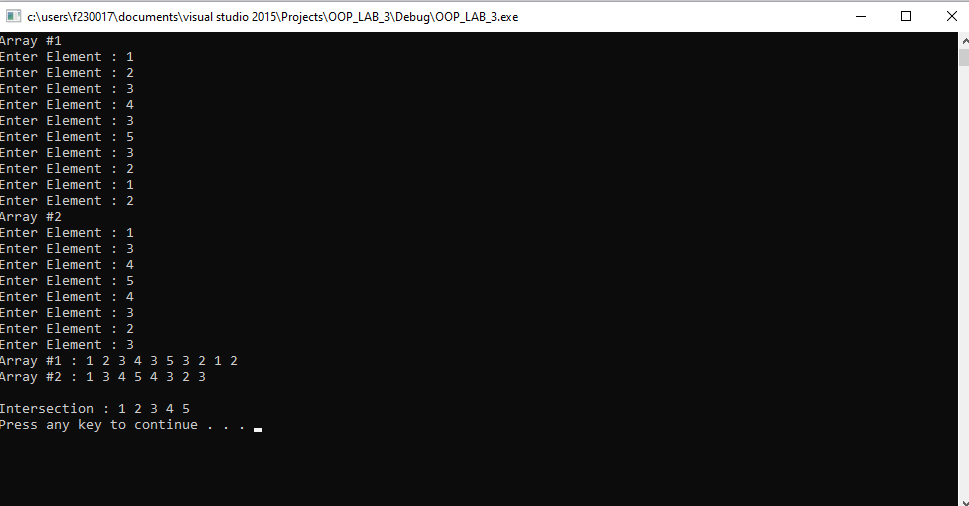
delete[] final;

system("pause");

return 0;

}

Output:



Task#4

Code:

#include<iostream>

using namespace std;

int\* InputArray(const int& size) {

int\* arr = new int[size];

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

cout << "Enter Element #" << i + 1 << " : "; cin >> \*(arr + i);

}

return arr;

}

int\* CompressArray(int\* orignalArr, int& size) {

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

for (int j = i + 1; j < size; j++) {

if (\*(orignalArr + i) == \*(orignalArr + j)) {

\*(orignalArr + j) = -999999;

}

}

}

int count = 0;

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

if (\*(orignalArr + i) > -999999) {

count = count + 1;

}

}

int\* newarr = new int[count];

int temp = 0;

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

if (\*(orignalArr + i) > -999999) {

\*(newarr + temp) = \*(orignalArr + i);

temp = temp + 1;

}

}

size = count;

return newarr;

}

void OutputArray(int\* const ptr, const int& size) {

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

cout << \*(ptr + i) << " ";

}

cout << endl;

}

int main() {

int size = 0;

cout << "Enter The Size of Array : "; cin >> size;

int\* ptr = InputArray(size);

int\* ptr\_1 = CompressArray(ptr, size);

cout << "Output : ";

OutputArray(ptr\_1, size);

delete[] ptr;

delete[] ptr\_1;

system("pause");

return 0;

}

Output:



Task#5

Code:

#include<iostream>

using namespace std;

char\* getString(int &n) {

char\*\* arr = new char\* [50];

for (int i = 0; i < 50; i++) {

arr[i] = new char[50];

}

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

cout << "Enter a Sentence : ";cin >> arr[i];

}

int count = 0;

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

for (int j = 0; j < 50; j++) {

count = count + 1;

if (arr[i][j] == '\0') {

break;

}

}

}

char\* arr1D = new char[count + 1];

int k = 0;

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

for (int j = 0; j < count; j++) {

arr1D[k] = arr[i][j];

if (arr1D[k] == '\0') {

break;

}

k = k + 1;

}

}

arr1D[count] = '\0';

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

delete[] arr[i];

}

delete[] arr;

return arr1D;

}

int main() {

int n;

cout << "Enter No. of Inputs : "; cin >> n;

char\* ptr = getString(n);

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

cout << ptr[i];

if (ptr[i] == '\0') {

break;

}

}

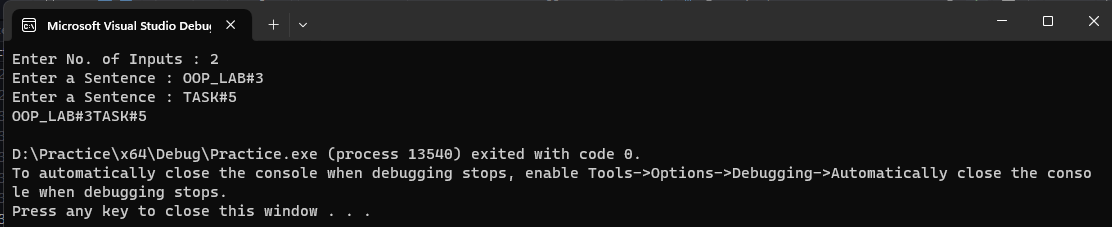
cout << endl;

delete[] ptr;

return 0;

}

Output:



Task#6

Code:

#include<iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int\* copyFun(int arr[], int size) {

int\* newArr = new int[size];

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

\*(newArr + i) = arr[i];

}

int\* ptr = newArr;

return ptr;

}

int main() {

srand(time(0));

const int size = 10;

int arr[size];

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

arr[i] = rand() % 101;

}

cout << "Original Array : ";

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

cout << arr[i] << " ";

}

cout << endl;

int \*ptr =copyFun(arr, size);

cout << "Duplicate Array : ";

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

cout << \*(ptr + i) << " ";

}

cout << endl;

delete[] ptr;

return 0;

}

Output:

A screen shot of a computer

Description automatically generated