**Question#1**

**Code:**

#include<iostream>

using namespace std;

int main() {

double var1, var2;

cout << "Enter Value #1 : "; cin >> var1;

cout << "Enter Value #2 : "; cin >> var2;

double\* ptrV1 = &var1;

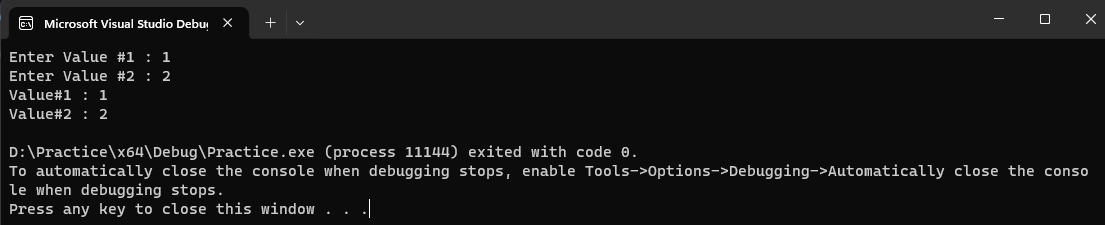
double\* ptrV2 = &var2;

cout << "Value#1 : " << \*ptrV1 << endl;

cout << "Value#2 : " << \*ptrV2 << endl;

return 0;

}

**Output:** ****

**Question#2**

**Code:**

#include<iostream>

#include<cstring>

using namespace std;

void rollno(int &a,int &b) {

char arr[5];

int arr1[4];

cout << "Enter Last Four Digits of Roll No : "; cin >> arr;

for (int i = 0; i < 4; i++) {

arr1[i] = arr[i] - '0';

}

a = arr1[0] + arr1[1] + 1;

b = arr1[2] + arr1[3] + 1;

cout << "a = " << a << ", b = " << b << endl;

}

void GenerateCode(int\* P1, int\* P2, char\* line) {

for (int i = 0; i < strlen(line); i++) {

if (line[i] >= 65 && line[i] <= 90) {//uppercase

for (int j = 0; j < (\*P1); j++) {

if (line[i] + 1 > 90) {

line[i] = 'A' +1;

}

else {

line[i] = line[i] + 1;

}

}

}

else if (line[i] >= 97 && line[i] <= 122) {

//lower case

for (int j = 0; j < (\*P2); j++) {

if (line[i] + 1> 122) {

line[i] = 'A' + 1;

}

else {

line[i] = line[i] + 1;

}

}

}

}

}

void Decode(int\* P1, int\* P2, char\* line) {

for (int i = 0; i < strlen(line); i++) {

if (line[i] >= 65 && line[i] <= 90) {//uppercase

for (int j = 0; j < (\*P1); j++) {

if (line[i] - 1 < 65) {

line[i] = 'Z' - 1;

}

else {

line[i] = line[i] - 1;

}

}

}

else if (line[i] >= 97 && line[i] <= 122) {

//lower case

for (int j = 0; j < (\*P2); j++) {

if (line[i] - 1 < 97) {

line[i] = 'Z' - 1;

}

else {

line[i] = line[i] - 1;

}

}

}

}

}

int main() {

int a , b;

rollno(a, b);

int\* ptr1 = &a;

int\* ptr2 = &b;

char line[100];

cin.ignore();

cout << "Enter Line : "; cin.getline(line, 100);

GenerateCode(ptr1, ptr2, line);

cout << "Encrypted Line : ";

for (int i = 0; i < strlen(line); i++) {

cout << line[i];

}

cout << endl;

Decode(ptr1,ptr2,line);

cout << "Decrypted Line : ";

for (int i = 0; i < strlen(line); i++) {

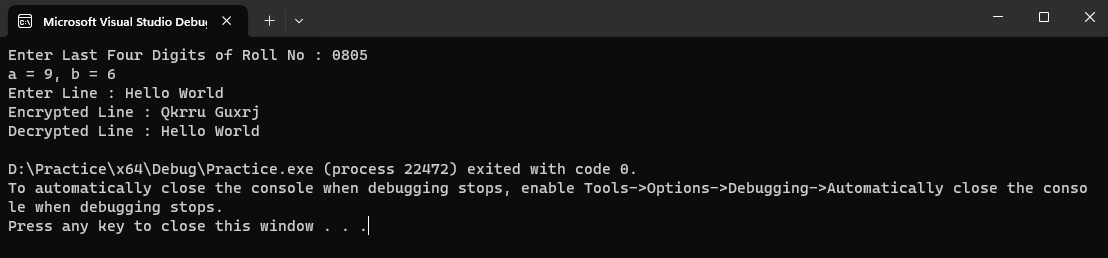
cout << line[i];

}

cout << endl;

return 0;

}

**Output:** ****

**Question#3**

**Code:**

#include<iostream>

using namespace std;

int strlen(char\* str) {

int i;

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

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

break;

}

}

return i;

}

int strcmp(const char \*str, const char \*str1) {

bool same = true;

if (strlen(str) == strlen(str1)) {

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

if (str[i] != str1[i]) {

same = false;

}

else {

same = true;

}

}

if (same == true) {

return 0;

}

}

else {

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

if (str[i] > str1[i]) {

return 1;

break;

}

else if (str[i] < str1[i]) {

return -1;

break;

}

}

}

}

int strncmp(const char\* str, const char\* str1,int n){

bool same = true;

if (strlen(str) == strlen(str1)) {

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

if (str[i] != str1[i]) {

same = false;

}

else {

same = true;

}

}

if (same == true) {

return 0;

}

}

else {

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

if (str[i] > str1[i]) {

return 1;

break;

}

else if (str[i] < str1[i]) {

return -1;

break;

}

}

}

}

char\* strcpy(char\* str, const char\* str1) {

for (int i = 0; i < strlen(str1); i++) {

str[i] = str1[i];

}

return str;

}

char\* strncpy(char\* str, const char\* str1, int n) {

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

str[i] = str1[i];

}

return str;

}

char\* strcat(char\* str,const char\* str1) {

int j = strlen(str);

for (int i = 0; i < strlen(str1); i++) {

str[j + i] = str1[i];

}

return str;

}

char\* strncat(char\* str, const char\* str1, int n) {

int j = strlen(str);

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

str[j + i] = str1[i];

}

return str;

}

int main() {

char str[100] = { "hello" };

char str1[100] = { "world" };

int n = 5;

cout << "String Length Is : " << strlen(str) << endl;

cout << "strcmp(str,str1) : " << strcmp(str, str1) << endl;

cout << "strncmp(str,str1,n) : " << strncmp(str, str1,n) << endl;

cout << "strcat(str,str1) : " << strcat(str, str1) << endl;

n = 3;

cout << "strncat(str,str1,n) : " << strncat(str, str1, n) << endl;

return 0;

}

**Output:**

**Question#4**

**Code:**

#include<iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

void addArray(int arr1[], int arr2[], int arr3[], const int size) {

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

\*(arr3 + i) = \*(arr1 + i) + \*(arr2 + i);

}

}

int main() {

srand(time(0));

const int size = 10;

int arr1[size] = { 0 };

int arr2[size] = { 0 };

int arr3[size] = { 0 };

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

\*(arr1 + i) = rand() % 10 + 1;

\*(arr2 + i) = rand() % 10 + 1;

}

cout << "Array#1 : ";

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

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

}

cout << endl;

cout << "Array#2 : ";

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

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

}

cout << endl;

addArray(arr1, arr2, arr3, size);

cout << "Array#3 : ";

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

cout << \*(arr3 + i) << " ";

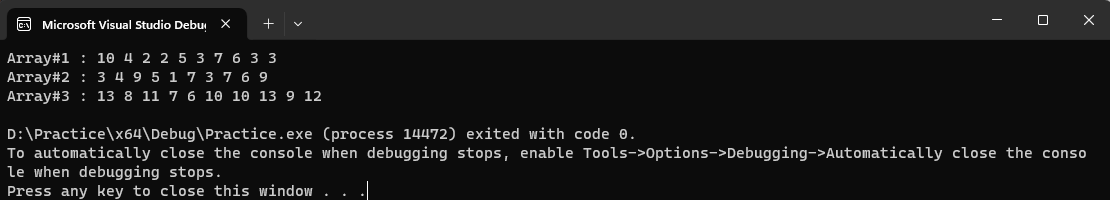
}

cout << endl;

return 0;

}

**Output:**

****

**Question#5**

**Code:**

#include<iostream>

#include<fstream>

using namespace std;

int\*\* InputMatrix(ifstream& fin, int& rows, int& cols) {

int\*\* matrix = new int\* [rows];

int \*i=new int;

int\* j=new int;

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

\*(matrix + \*i) = new int[cols];

}

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

for (\*j = 0; \*j < cols; (\*j = \*j + 1)) {

fin >> \*(\*(matrix + \*i) + \*j);

}

}

delete i;

delete j;

return matrix;

}

void OutputMatrix(int\*\* matrix, const int& rows, const int& cols) {

int\* i=new int;

int\* j=new int;

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

for (\*j = 0; \*j < cols; (\*j = \*j + 1)) {

cout << \*(\*(matrix + \*i) + \*j) << " ";

}

cout << endl;

}

delete i;

delete j;

}

int\*\* AddMatrix(int\*\* matrixA, int\*\* matrixB, const int& rows, const int& cols) {

int\*\* R = new int\* [rows];

int\* i=new int;

int\* j = new int;

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

\*(R + \*i) = new int[cols];

}

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

for (\*j = 0; \*j < cols; (\*j = \*j + 1)) {

\*(\*(R + \*i) + \*j) = \*(\*(matrixA + \*i) + \*j) + \*(\*(matrixB + \*i) + \*j);

}

}

delete i;

delete j;

return R;

}

int\*\* TransposeMatrix(int\*\* matrix, const int& rows, const int& cols) {

int\*\* newmatrix = new int\* [rows];

int\* i = new int;

int\* j = new int;

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

\*(newmatrix + \*i) = new int[cols];

}

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

for (\*j = 0; \*j < cols; (\*j = \*j + 1)) {

\*(\*(newmatrix + \*i) + \*j) = \*(\*(matrix+ \*j) + \*i);

}

}

delete i;

delete j;

return newmatrix;

}

bool isSymemetric(int\*\* matrix, const int& rows, const int& cols) {

int\*\* transpose = TransposeMatrix(matrix, rows, cols);

int\* i = new int;

int\* j = new int;

for (\*i = 0; \*i < rows; (\*i = \*i + 1)) {

for (\*j = 0; \*j < cols; (\*j = \*j + 1)) {

if (\*(\*(matrix + \*i) + \*j) != \*(\*(transpose + \*i) + \*j)) {

return false;

}

}

}

delete i;

delete j;

return true;

}

void InterChangeRows(int\*& row1, int\*& row2) {

int temp = \*row1;

\*row1 = \*row2;

\*row2 = temp;

}

void interChangeRows(int\*\* matrix, const int& rows, const int& cols) {

int r1 = 0, r2 = 0;

cout << "Enter Row Number To Interchange" << endl;

cout << "Enter : "; cin >> r1;

cout << "Enter : "; cin >> r2;

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

int\* temp = new int;

int\* temp1 = new int;

\*temp = \*(\*(matrix + r1 - 1) + i);

\*temp1 = \*(\*(matrix + r2 - 1) + i);

InterChangeRows(temp,temp1);

\*(\*(matrix + r1 - 1) + i) = \*temp;

\*(\*(matrix + r2 - 1) + i) = \*temp1;

delete temp;

delete temp1;

}

}

int main() {

int rows = 0, cols = 0;

ifstream file;

file.open("info.txt", ios::in);

file >> rows;

file >> cols;

int \*\*matrixA = InputMatrix(file, rows, cols);

int\*\* matrixB = InputMatrix(file, rows, cols);

cout << "Matrix A : " << endl;

OutputMatrix(matrixA, rows, cols);

cout << "Matrix B : " << endl;

OutputMatrix(matrixB, rows, cols);

int\*\* result = AddMatrix(matrixA, matrixB, rows, cols);

cout << "Resultant Matrix A + B : " << endl;

OutputMatrix(result, rows, cols);

cout << "Transpose of Matrix A : " << endl;

int\*\* transpose = TransposeMatrix(matrixA, rows, cols);

OutputMatrix(transpose, rows, cols);

interChangeRows(matrixA, rows, cols);

cout << "Interchanged Matrix A : " << endl;

OutputMatrix(matrixA, rows, cols);

file.close();

return 0;

}

**Output: A screenshot of a computer

Description automatically generated**

**Question#6**

**Code:**

#include<iostream>

#include<cstdlib>

#include<ctime>

#include<Windows.h>

using namespace std;

void moveTortoise(int\* tortoise) {

int move = rand() % 50 + 1;

if (move <= 50 && move > 35) {//slow plod

\*tortoise = \*tortoise + 1;

}

else if (move <= 35 && move > 25) {//slip

\*tortoise = \*tortoise + 5;

}

else if (move < 25 && move>1) {//fat plod

\*tortoise = \*tortoise + 4;

}

}

void moveHare(int\* hare) {

int move = rand() % 50 + 1;

if (move <= 50 && move > 35) {//small hop

\*hare = \*hare + 1;

}

else if (move <= 35 && move > 25) {//big hop

\*hare = \*hare + 11;

}

else if (move < 25 && move>1) {//slip

\*hare = \*hare - 9;

if (\*hare <= 0) {

\*hare = 1;

}

}

}

int main() {

srand(time(0));

int hare = 0;

int tortoise = 0;

for (int i = 0; i < 70; i++) {

moveTortoise(&tortoise);

moveHare(&hare);

for (int i = 0; i < 70; i++) {

cout << "-";

}

cout << endl;

if (tortoise >= 70 && hare >= 70) {

cout << "Its a Tie" << endl;

}

if (tortoise >= 70) {

cout << "Tortoise Won The Race" << endl;

break;

}

if (hare >= 70) {

cout << "Hare Won The Race" << endl;

break;

}

for (int j = 1; j <= tortoise; j++) {

if (j == tortoise) {

if (hare == tortoise) {

cout << "Ouch!!!" << endl;

}

else {

cout << "T" << endl;

}

}

else {

cout << " ";

}

}

cout << endl;

for (int j = 1; j <= hare; j++) {

if (j == hare) {

if (hare == tortoise) {

cout << "Ouch!!!" << endl;

}

else {

cout << "H" << endl;

}

}

else {

cout << " ";

}

}

for (int i = 0; i < 70; i++) {

cout << "-";

}

cout << endl;

Sleep(500);

system("cls");

}

return 0;

}

**Output:**

A screen shot of a computer

Description automatically generated