Experiment #8

1.

b)

#include<iostream>

using namespace std;

void swap1(int a, int b) {

int temp = a;

a = b;

b = temp;

}

void swap2(int a, int b) {

int\* temp = &a;

a = b;

b = \*temp;

}

void swap3(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

void swap4(int\* a, int\* b) {

int temp0 = 0;

int\* temp = &temp0;

\*temp = \*a;

\*a = \*b;

\*b = \*temp;

}

void swap5(int& a, int& b) {

int temp = a;

a = b;

b = temp;

}

int main() {

int a = 1, b = 9;

swap1(a, b);

cout << "a = " << a << '\t' << "b = " << b << endl;

a = 1, b = 9;

swap2(a, b);

cout << "a = " << a << '\t' << "b = " << b << endl;

a = 1, b = 9;

swap3(&a, &b);

cout << "a = " << a << '\t' << "b = " << b << endl;

a = 1, b = 9;

swap4(&a, &b);

cout << "a = " << a << '\t' << "b = " << b << endl;

a = 1, b = 9;

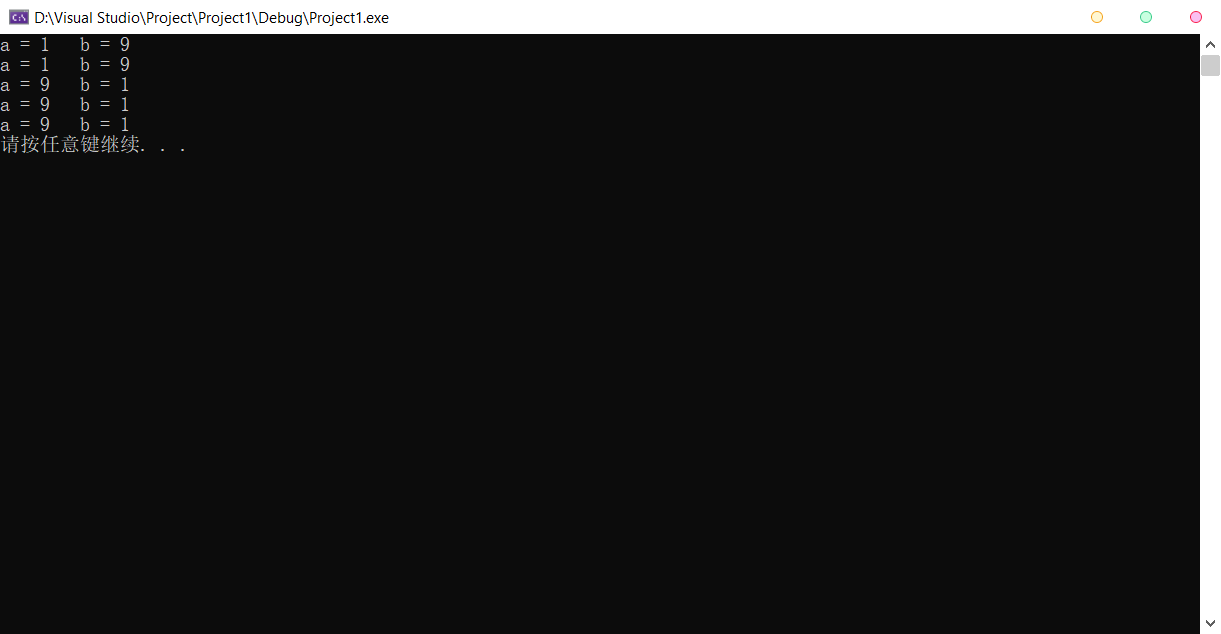
swap5(a, b);

cout << "a = " << a << '\t' << "b = " << b << endl;

system("pause");

return 0;

}



c)

#include<iostream>

using namespace std;

int getsum1(int a, int b) {

int temp = a + b;

return temp;

}

int\* getsum2(int a, int b) {

int temp = a + b;

return &temp;

}

int\* getsum3(int a, int b) {

int temp0 = a + b;

int\* temp = &temp0;

\*temp = a + b;

return temp;

}

int getsum4(int a, int b) {

int temp0 = a + b;

int\* temp = &temp0;

return \*temp;

}

int getsum5(int a, int b) {

int temp0 = a + b;

int& temp = temp0;

return temp;

}

int main() {

int a = 1, b = 9;

cout << "1.return temp\n" << getsum1(a, b) << endl << endl;

cout << "2.return the address of temp\n" << getsum2(a, b) << endl << endl;

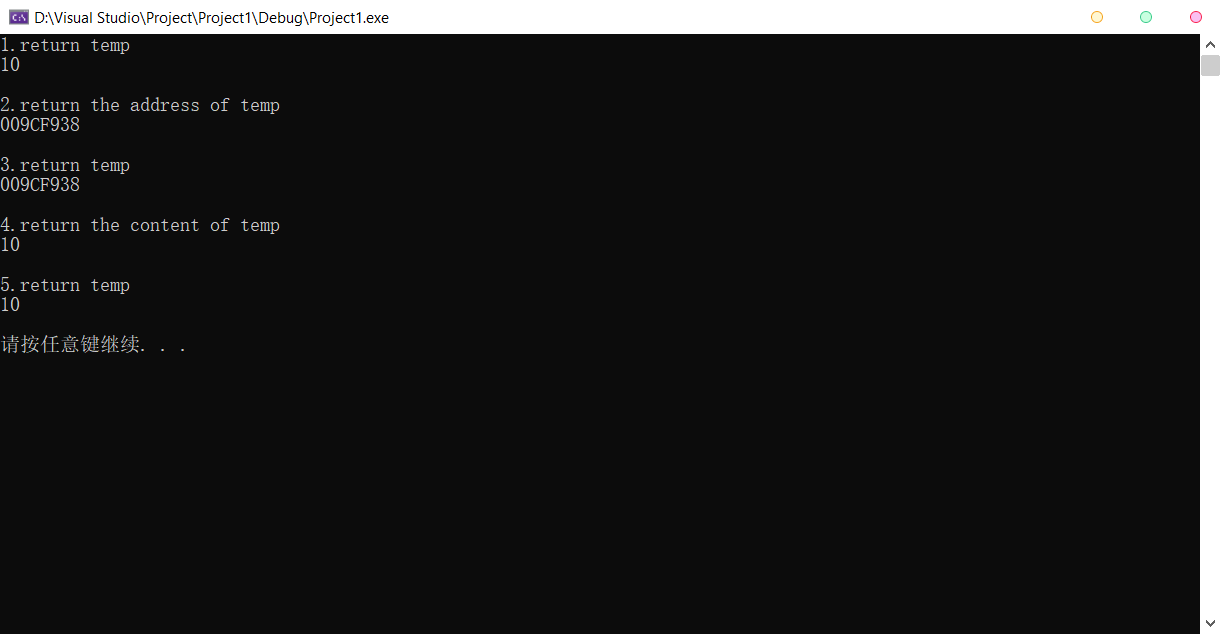
cout << "3.return temp\n" << getsum3(a, b) << endl << endl;

cout << "4.return the content of temp\n" << getsum4(a, b) << endl << endl;

cout << "5.return temp\n" << getsum5(a, b) << endl << endl;

system("pause");

return 0;

}

2.

#include<iostream>

using namespace std;

int main() {

const int r = 3, c = 5;

int m[r][c] = { 0 };

int(\* p)[c] = m;

cout << "Input a 3\*5 matrix:\n";

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

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

cin >> m[i][j];

}

}

cout << "\nThe matrix you entered is:\n";

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

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

cout << m[i][j] << '\t';

}

cout << endl;

}

cout << "\nThe matrix after transposed is:\n";

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

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

cout << p[j][i] << '\t';

}

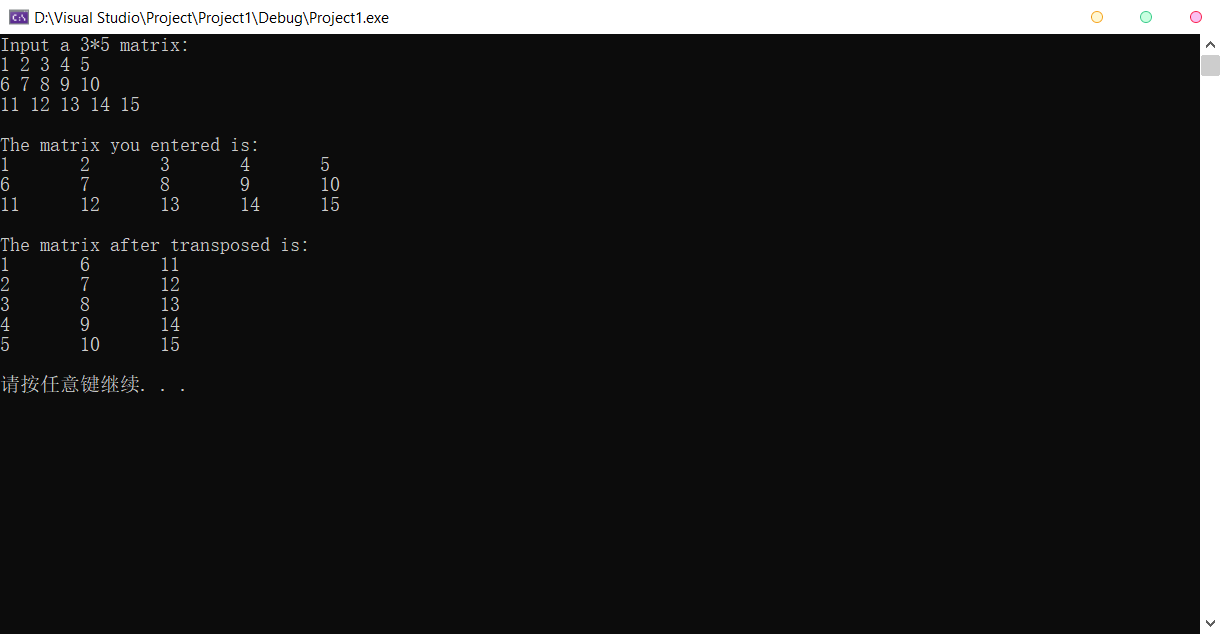
cout << endl;

}

cout << endl;

system("pause");

return 0;

}

3.

#include <iostream>

using namespace std;

int main() {

const int size = 100;

char s1[size] = { 0 }, s2[size] = { 0 };

char\* p = s1;

cout << "Please input two strings:\n";

cin.getline(s1, size);

cin.getline(s2, size);

for (; \*p != '\0'; p++) {

if (\*p >= 97 && \*p <= 122)

\*p -= 32;

}

\*--p = ',';

p = s2;

for (; \*p != '\0'; p++) {

if (\*p >= 97 && \*p <= 122)

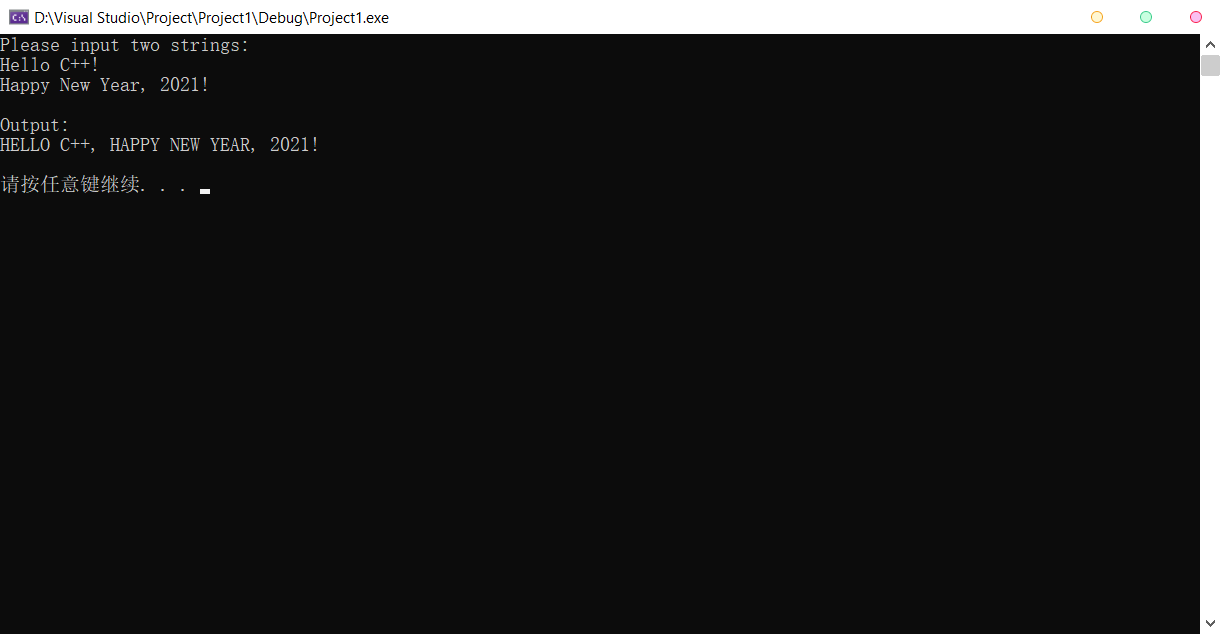
\*p -= 32;

}

cout << "\nOutput:\n" << s1 << ' ' << s2 << endl << endl;

system("pause");

return 0;

} 

4.

#include<iostream>

using namespace std;

const int ArraySize = 50;

bool paren\_check(char\* expr) {

int lp = 0, rp = 0;

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

if ( expr[i] == '(')

lp++;

if (expr[i] == ')')

rp++;

}

if (lp + rp == 0)

cout << "NO parentheses!\n";

return ((lp == rp) ? 1 : 0);

}

int main() {

char expr[ArraySize] = { 0 };

cout << "Enter an expression:\n";

cin.getline(expr, ArraySize);

cout << paren\_check(expr) << endl;

system("pause");

return 0;

}