Лаб 3

№2

#include<iostream>

#include<string>

using namespace std;

const int n = 5;

int Max(int a, int b);

int Max(int x, int y, int z);

double Max(double u, double v);

string Max(string str, string str1);

int Max(int \*A, int n);

char Max(char B[], int n);

void main()

{

int a = 12, b = 8, x = 3, y = 6, z = -12;

double u = 31.2, v = 24.5;

string str = "C++", str1 = "C++";

int A[n] = { 2, 5, 1, -4, 11 };

char B[n] = "Word";

cout << Max(a, b) << endl;

cout << Max(x, y, z) << endl;

cout << Max(u, v) << endl;

cout << "Max is: " << Max(str, str1) << endl;

cout << Max(A, n) << endl;

cout << Max(B, n) << endl;

cin.get();

cin.get();

}

int Max(int a, int b) {

if (a > b)

return a;

else

return b;

if (a == b)

return a;

}

int Max(int x, int y, int z) {

if (x > y)

{

if (x > z)

return x;

}

else

if (x < y)

{

if (y>z)

return y;

else

if (y < z)

return z;

}

}

double Max(double u, double v) {

if (u>v)

return u;

else

return v;

}

string Max(string str, string str1)

{

if (str.size() == str1.size())

return str.append(str1);

if (str.size() > str1.size())

return str;

else

return str1;

}

int Max(int \*A, int n) {

int max = A[0];

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

{

if (A[i]>max)

max = A[i];

}

return max;

}

char Max(char B[], int n) {

char max = B[0];

for (int i = 0; i < strlen(B); i++)

{

if (B[i]>max)

max = B[i];

}

return max;

}

№3

#include<iostream>

using namespace std;

const int n = 3, m = 4;

struct My\_struct {

int number;

}first[n], second[m];

void Merge(int A[], int B[], int n, int m);

void Merge(double \*X, double \*Y, int n, int m);

void Merge(My\_struct first[], My\_struct second[], int n, int m);

void Merge(char string1[], char string2[],int n,int m);

void main()

{

int A[n] = { 1,6,8 }, B[m] = { 2,5,9,11 };

double X[n] , Y[m] = { 2.33,4.5,6.0,11.2 };

cout << "Please,enter elements of array X: ";

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

cin >> X[i];

cout << "Enter elements of array of struct FIRST: ";

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

cin >> first[i].number;

cout << endl;

cout << "Enter elements of array of struct SECOND: ";

for (int j(0); j < m;j++)

cin >> second[j].number;

cout << endl;

char string1[n] = { 'm','e','\0' }, string2[m] = "cat";

system("cls");

Merge(A, B, n, m);

cout << endl;

Merge(X, Y, n, m);

cout << endl;

Merge(first, second, n, m);

cout << endl;

Merge(string1, string2, n, m);

cin.get();

cin.get();

}

void Merge(int A[], int B[], int n, int m) {

int i = 0, j = 0, k = 0;

int \*C = new int[n + m];

while (i < n && j < m)

{

if (A[i] < B[j])

{

C[k] = A[i];

i++;

}

else {

C[k] = B[j];

j++;

}

k++;

}

if (i < n) {

while (k < n + m)

{

C[k] = A[i];

k++;

i++;

}

}

else

while (k < n + m)

{

C[k] = B[j];

k++;

j++;

}

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

cout<< C[i]<< '\t' ;

}

void Merge(double \*X, double \*Y, int n, int m) {

int i = 0, j = 0, k = 0;

double \*Z = new double[n + m];

while (i < n && j < m)

{

if (X[i] < Y[j])

{

Z[k] = X[i];

i++;

}

else {

Z[k] = Y[j];

j++;

}

k++;

}

if (i < n) {

while (k < n + m)

{

Z[k] = X[i];

k++;

i++;

}

}

else

while (k < n + m)

{

Z[k] = Y[j];

k++;

j++;

}

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

cout << Z[i]<<'\t';

}

void Merge(My\_struct first[], My\_struct second[], int n, int m) {

int i = 0, j = 0, k = 0;

My\_struct \*third = new My\_struct[n + m];

while (i < n && j < m)

{

if (first[i].number < second[j].number)

{

third[k].number = first[i].number;

i++;

}

else {

third[k].number = second[j].number;

j++;

}

k++;

}

if (i < n) {

while (k < n + m)

{

third[k].number = first[i].number;

k++;

i++;

}

}

else

while (k < n + m)

{

third[k].number = second[j].number;

k++;

j++;

}

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

cout << third[i].number << '\t';

}

void Merge(char \*string1, char \*string2, int n, int m) {

int i = 0, j = 0, k = 0;

char \*string3 = new char[n + m];

while (i < n && j < m)

{

if ((int)string1[i] < (int)string2[j])

{

string3[k] = string1[i];

i++;

}

else {

string3[k] = string2[j];

j++;

}

k++;

}

if (i < n) {

while (k < n + m)

{

string3[k] = string1[i];

k++;

i++;

}

}

else

while (k < n + m)

{

string3[k] = string2[j];

k++;

j++;

}

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

cout << string3[i] << '\t';

}

#include <iostream>

using namespace std;

template <typename T>

void swa(T &x, T &y)

{

T tmp = x;

x = y;

y = tmp;

cout << "x=" << x<<endl;

cout << "y=" << y << endl;

}

int main()

{

int i = 10, j = 20;

double x = 10.1, y = 20.2;

char c = 'x', d = 'z';

swa(i, j);

swa(x, y);

swa(c, d);

cin.get();

return 0;

}