МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

УНІВЕРСИТЕТ БАНКІВСЬКОЇ СПРАВИ

Лабораторна робота №4

з дисципліни : Об’єктно-орієнтоване програмування

Виконала студентка

1 курсу (скороченого терміну навчання)

Сорока Олена Олександрівна

КИЇВ 2020

**Lab 4.1.1 Multi-dimensional arrays of variable length**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{

float n, cu, gr, sum = 0, sumf = 0;

double fin;

cout << "How much Courses? ";

cin >> n;

cout << endl;

if (n <= 5)

{

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

{

cout << "Courses " << i + 1 << " grades: ";

cin >> cu;

if (cu <= 10)

{

cout << "Enter " << cu << " marks:" << endl;

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

{

cin >> gr;

if (gr <= 5)

sum += gr;

else

{

cout << "Error, try again" << endl;

break;

}

}

fin = sum / cu;

cout << "Final: " << fin << endl;

sum = 0;

sumf += fin;

}

else

{

cout << "Error, try again" << endl;

break;

}

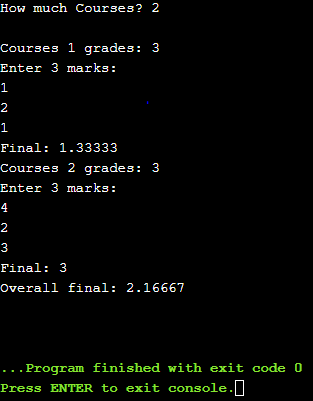
}

cout << "Overall final: " << sumf / n << "\n\n";

}

return 0;

}

****

**Lab 4.2.1 Multi** **Range of integer types**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{

long int n, summ = 0;

cout << "Enter n: ";

cin >> n;

if (n >= 1 && n <= 1000000)

{

for (int i = 1; i < n + 1; i++)

summ = summ + i;

cout << summ << endl;

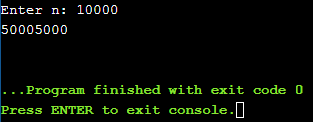
}

else

cout << "Error" << endl;

return 0;

**}**

****

**Lab 4.2.2 Comparing floating-point numbers**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

bool is\_close(double a, double b, double tolerance)

{

if (b >= a)

{

double k = b - a;

if (k <= tolerance)

return true;

else

return false;

}

else

{

double k = a - b;

if (k <= tolerance)

return true;

else

return false;

}

return false;

}

int main()

{

if (0.3 == 3 \* 0.1)

cout << "The numbers are equal";

else

cout << "The numbers are not equal";

cout << endl;

if (is\_close(0.3, 3 \* 0.1, 0.00000001))

cout << "The numbers are close enough";

else

cout << "The numbers are not close enough";

cout << endl;

if (is\_close(3 \* 0.1, 0.3, 0.00000001))

cout << "The numbers are still close enough";

else

cout << "The numbers are not close enough";

cout << endl;

if (is\_close(3 \* 0.1, 0.31, 0.00000001))

cout << "The numbers are still close enough";

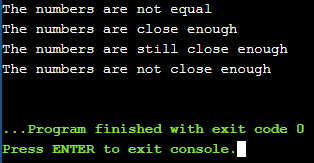
else

cout << "The numbers are not close enough";

cout << endl;

return 0;

}

****

**Lab 4.4.1 String manipulation: pangrams**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{

string str;

string alphabet = "ABCDEFGHIJKLMNOPQRSTVWXYZ";

cout << "Enter text: ";

getline(cin, str);

for (int j = 0; j < alphabet.size(); j++)

{

int count = 0;

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

{

if (tolower(str[i]) == tolower(alphabet[j]))

count++;

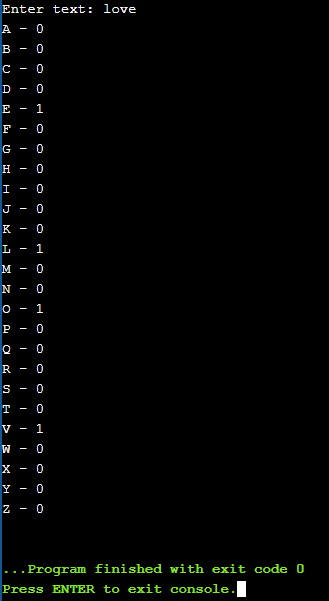
}

cout << alphabet[j] << " - " << count << endl;

}

return 0;

}



**Lab 4.5.1 String manipulation: pangrams**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{

char IP[16];

char\* endptr;

long int a;

cout << "Enter IP: ";

cin >> IP;

if (a = strtol(IP, &endptr, 10))

if (a >= 0 && a <= 255)

{

int i;

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

{

++endptr;

if (!(a = strtol(endptr, &endptr, 10)) || !(a >= 0 && a <= 255))

{

cout << "String is not IP";

break;

}

}

if (i < 3)

cout << "IP isn't correct";

else

cout << "Correct IP";

}

else

cout << "IP isn't correct";

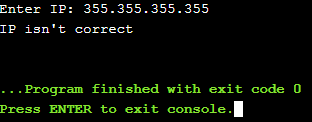
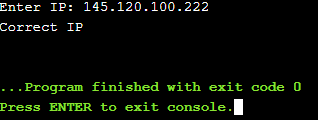
else

cout << "IP isn't only numbers";

cout << endl;

return 0;

}

** **

**Lab 4.5.2 Text manipulation: duplicate white space**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

int main()

{

string sentence;

cout << "Enter sentence: ";

getline(cin, sentence);

for (int i = 0; i < sentence.length(); i++)

if (sentence[i] == ' ' && sentence[i + 1] == ' ')

{

sentence.erase(i, 1);

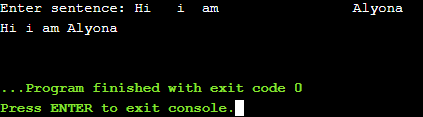
i = (i - 1);

}

cout << sentence << endl;

return 0;

}



**Lab 4.5.3 Text manipulation: eliminate duplicates**

**Code:**

#include <iostream>

#include<cstdlib>

#include<ctime>

#include <vector>

#include <sstream>

using namespace std;

int main()

{

string input;

cout << "Enter the sentence: ";

getline(cin, input);

string temp, output;

vector<string> vec;

stringstream ss;

ss << input;

while (ss >> temp)

vec.push\_back(temp);

int i;

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

{

if (vec[i] != vec[i + 1])

output += vec[i] + " ";

}

output += vec[i];

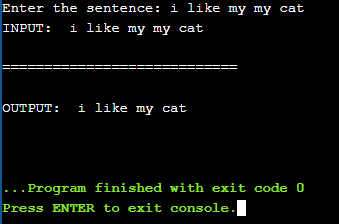
cout << "INPUT: " << input << endl;

cout << "\n============================\n\n";

cout << "OUTPUT: " << output << endl << endl;

return 0;

}

****

**Lab 4.5.4 Text manipulation: find and replace**

**Code:**

#include <iostream>

#include<cstdlib>

using namespace std;

int main()

{

string sentence;

cout << "Enter a string:" ;

getline(cin, sentence);

string from;

cout << "Enter a word you want to change:" ;

getline(cin, from);

string to;

cout << "Enter a new word:";

getline(cin, to);

int t = sentence.find(from);

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

cout << sentence[i];

cout << to;

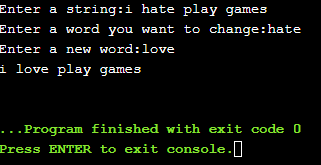
for (int i = to.length() + t; i < sentence.length(); i++)

cout << sentence[i];

cout << endl;

return 0;

}

****

**Lab 4.5.5 Text manipulation: anagrams**

**Code:**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

int main()

{

string a, b;

cout << "Enter two words: " << endl;

cin >> a >> b;

if (a.length() == b.length() && a.length() <= 20)

{

sort(a.begin(), a.end());

sort(b.begin(), b.end());

if (a == b)

cout << "Words are anagrams" << endl;

else

cout << "Words aren`t anagrams" << endl;

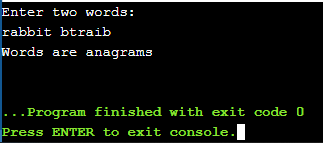
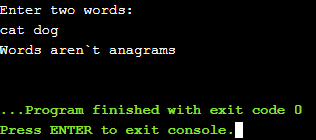
}

else

cout << "Error" << endl;

return 0;

}

****

**Lab 4.5.6 Text manipulation: pattern matching**

**Code:**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

int main()

{

string password;

cout << "Enter a password:";

getline(cin, password);

char uc\_letters[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

char letters[] = "abcdefghijklmnopqrstuvwxyz";

char digits[] = "1234567890";

char symbols[] = "/<>,.$%[]{}^&\*!@#\_-+=;:?";

if (password.length() < 8)

cout << "\nThe password must be 8 characters long!\n";

int k = 0;

for (int i = 0; i < 26; i++)

if (password.find(uc\_letters[i]) != string::npos)

k++;

if (k == 0)

cout << "\nThe password must have at least one upper case letter!" << endl;

int x = 0;

for (int i = 0; i < 26; i++)

if (password.find(letters[i]) != string::npos)

x++;

if (x == 0)

cout << "\nThe password must have at least one lower case letter!" << endl;

int y = 0;

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

if (password.find(digits[i]) != string::npos)

y++;

if (y == 0)

cout << "\nThe password must have at least one digit!" << endl;

int z = 0;

for (int i = 0; i < 24; i++)

if (password.find(symbols[i]) != string::npos)

z++;

if (z == 0)

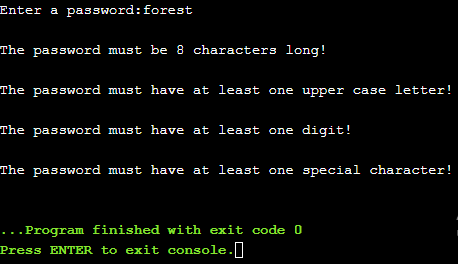
cout << "\nThe password must have at least one special character!" << endl;

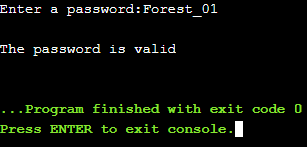
if ((password.length()) >= 8 && k != 0 && x != 0 && y != 0 && z != 0)

cout << "\nThe password is valid" << endl;

return 0;

}

****

****

**Lab 4.5.8 Text manipulation: templates**

**Code:**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

int main()

{

cin.ignore();

string data1, data2, data3, sen;

cout << "Enter data types we will work with:" << endl;

getline(cin, data1);

getline(cin, data2);

getline(cin, data3);

if (data1.find('[') == string::npos || data2.find('[') == string::npos || data3.find('[') == string::npos || data1.find(']') == string::npos || data2.find(']') == string::npos || data3.find(']') == string::npos)

cout << "Wrong data types" << endl;

else

{

cout << "Enter the sentence with all data types:" << endl;

getline(cin, sen);

if (sen.find(data1) == string::npos || sen.find(data2) == string::npos || sen.find(data3) == string::npos)

cout << "One or more of data types wasn`t used" << endl;

else

{

string d1, d2, d3;

cout << "Enter the data:" << endl;

getline(cin, d1);

getline(cin, d2);

getline(cin, d3);

sen.replace(sen.find('['), data1.length(), d1);

sen.replace(sen.find('['), data2.length(), d2);

sen.replace(sen.find('['), data3.length(), d3);

cout << sen << endl;

}

}

return 0;

}