**Задание для кандидата в программисты для студентов.**

**Теоретическая часть**

Тут представлены 3 вопроса теоретического характера, где нужно изложить ответ в письменном виде.

**Вопрос 1:**

Найдите ошибки в коде:

int main(int argc, char\*\* argv) {

int array = new int(30);//синтаксическая ошибка нет знака \* перед array, скобки заменить на угловые [ ] .правильно-int \*array = new int[30]

for (size\_t i = 0; i < 30; ++i) {

array[i] = i;

}

return 0;

}

**Вопрос 2:**

Найдите ошибки в коде и объясните:

class A {

public:

void f1() {

++a;

}

void f2() {

++b;

}

private:

int a;//не объявлена переменная

static int b;

};

int A::b = 0x7FFF;

int main(int argc, char\*\* argv) {

A\* b;//не выделена память под указатель A \*b=new A;

b->f2();//

b->f1();

delete b;

return 0;

}

**Вопрос 3:**

Почему невозможно вызвать функцию f()? Предложите различные решения, чтобы можно было вызвать её.

void f(int a = 0) {

//some code

}

void f() {

//some code

}

**1)убрать optional parametr void f(int a = 0) сделать функцию с параметром int a void f(int a) . Т.к. компилятор из-за необязательного парметра считает две функцию за одну и не может однозначно определить какую использовать**

**2)написать вместо двух функций одну функцию с необязательным параметром void f(int a = 0) {**

//some code

**}**

**Вопрос 4:**

Какое максимальное и минимальное значение может вернуть функция?

int someFunc() {

srand(time(NULL));

int result = 0;

for(int i = 0; i < 100; ++i) {

int result += rand() % 2;

}

return result;

}

**максимальное и минимальное значение равно 0;**

**Практическая часть**

Тут представлены условия 5 задач для практического выполнения. Нужно представить решение в виде кода на языке С++.

**Задание 1:**

Программа ждет ввода строки неограниченной длины.

Затем программа ждет ввода символа.

После чего она должна разбить первую строку в другие подстроки, используя второй параметр в виде разделителя и вывести на экран.

#include <iostream>

#include <iostream>

#include <string.h>

using namespace std;

int main() {

string input;

char sep;

int i = 0;

int j = 0;

cin >> input;

cin >> sep;

while (i <= input.length()) {

if (input[i] == sep || i == input.length()) {

char \*bufstr = new char[i - j];

memcpy(&bufstr[0], &input[j], i - j);

bufstr[i - j] = '\0';

cout << bufstr << endl;

j = i + 1;

}

i++;

}

return 0;

}

**Задание 2:**

Заполните случайными значениями два массива A и B. Выведите на экран сначала A потом B. Объедините A и B в массив C, отсортируйте его по возрастанию и выведите его на экран. Выведите максимальное, минимальное и среднее геометрическое значения из массива C.

#include <iostream>

#include <math.h>

using namespace std;

class Massive {

private:

unsigned int N;

int \*array;

public:

Massive(unsigned int size) {

N = size;

array = new int[N - 1];

srand(time(NULL));

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

array[i] = rand() % N;

}

}

~Massive() {

delete[] array;

}

void print() {

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

cout << array[i] << ' ';

}

cout << endl;

}

void sort() {

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

for (int j = 0; j < N - 1; j++) {

if (array[j] > array[j + 1]) {

int b = array[j];

array[j] = array[j + 1];

array[j + 1] = b;

}

}

}

}

Massive &operator+(Massive &obj) {

Massive \*res = new Massive(this->N + obj.N);

int i = 0;

for (i; i < N; i++) {

res->array[i] = array[i];

}

for (i; i < res->N; i++) {

res->array[i] = obj.array[i - this->N];

}

return \*res;

}

Massive &operator=(Massive &obj) {

Massive \*res = new Massive(obj.N + 1);

this->N = obj.N;

for (int i = 0; i < res->N; i++) {

res->array[i] = obj.array[i];

this->array[i] = obj.array[i];

}

return \*res;

}

int Max\_val() {

int max = array[0];

for (int i = 1; i < N; i++) {

if (array[i] > max) {

max = array[i];

}

}

return max;

}

int Min\_val() {

int min = array[0];

for (int i = 1; i < N; i++) {

if (array[i] < min) {

min = array[i];

}

}

return min;

}

float Mid\_val() {

float mid = 1;

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

mid = mid \* array[i];

}

mid = pow(mid, (float)1 / N);

}

};

int main() {

Massive A(10);

Massive B(12);

A.print();

B.print();

Massive C = A + B;

C.print();

C.sort();

C.print();

cout << "Max val=" << C.Max\_val() << endl;

cout << "Min val=" << C.Min\_val() << endl;

cout << "Mid val=" << C.Mid\_val() << endl;

return 0;

}

**Задание 3:**

*Необходимо реализовать без использования регулярных выражений*

У вас есть vector<string> содержащий url.

Например:

std::vector<std::string> urls;

urls.push\_back("http://mytona.com/index.html");

urls.push\_back("https://mytona.com:8080/");

urls.push\_back("mytona.com");

Необходимо вывести на экран информацию о каждом url в формате

Protocol: http Host: mytona.com Port: 80 Page: /index.html

Если протокол не найден, то по умолчанию "http".

Если порт не найден, то по умолчанию "80"

Если страница не найдена, то по умолчанию "/"

#include <iostream>

#include <vector>

#include "string.h"

using namespace std;

void treatment(vector<string> &urls) {

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

string strbuf = urls.at(i);

int count\_slesh = 0;

int count\_two\_points = 0;

for (int j = 0; j < strbuf.length(); j++) {

if (strbuf[j] == '/') { count\_slesh++; }

if (strbuf[j] == ':') { count\_two\_points++; }

}

switch (count\_slesh) {

case 0: {

if (count\_two\_points == 1) {

int pos\_point = strbuf.find(':');

urls.at(i) = "Protocol:http Host:" + strbuf.substr(0, pos\_point) + " Port"

+ strbuf.substr(pos\_point, strbuf.length() - pos\_point) + " Page:/";

} else { urls.at(i) = "Protocol:http Host:" + strbuf + " Port:80 Page:/"; }

break;

}

case 1: {

if (count\_two\_points == 1) {

int pos\_point = strbuf.find(':');

int pos\_slesh = strbuf.find('/');

urls.at(i) = "Protocol:http Host:" + strbuf.substr(0, pos\_point) + " Port"

+ strbuf.substr(pos\_point, pos\_slesh - pos\_point) + " Page:" +

strbuf.substr(pos\_slesh, strbuf.length() - pos\_slesh);

} else {

int pos\_slesh = strbuf.find('/');

urls.at(i) = "Protocol:http Host:" + strbuf.substr(0, pos\_slesh) + " Port:80 Page:/"

+ strbuf.substr(pos\_slesh + 1, strbuf.length());

}

break;

}

case 2: {

if (count\_two\_points == 2) {

int pos\_point = strbuf.find(':');

urls.at(i) = "Protocol:" + strbuf.substr(0, pos\_point) + " Host:";

strbuf = strbuf.substr(pos\_point + 3, strbuf.length());

pos\_point = strbuf.find(':');

urls.at(i) = urls.at(i) + strbuf.substr(0, pos\_point) + " Port" +

strbuf.substr(pos\_point, strbuf.length()) + " Page:/";

} else {

int pos\_point = strbuf.find(':');

urls.at(i) = "Protocol:" + strbuf.substr(0, pos\_point) + " Host:"

+ strbuf.substr(pos\_point + 3, strbuf.length()) + " Port:80 Page:/";

}

break;

}

case 3: {

if (count\_two\_points == 2) {

int pos\_point = strbuf.find(':');

urls.at(i) = "Protocol:" + strbuf.substr(0, pos\_point) + " Host:";

strbuf = strbuf.substr(pos\_point + 3, strbuf.length());

pos\_point = strbuf.find(':');

int pos\_slesh = strbuf.find('/');

urls.at(i) = urls.at(i) + strbuf.substr(0, pos\_point) + " Port" +

strbuf.substr(pos\_point, pos\_slesh - pos\_point)

+ " Page:"

+ strbuf.substr(pos\_slesh, strbuf.length());

} else {

int pos\_point = strbuf.find(':');

urls.at(i) = "Protocol:" + strbuf.substr(0, pos\_point) + " Host:";

strbuf = strbuf.substr(pos\_point + 3, strbuf.length());

int pos\_slesh = strbuf.find('/');

urls.at(i) = urls.at(i) + strbuf.substr(0, pos\_slesh) + " Port:80" + " Page:"

+ strbuf.substr(pos\_slesh, strbuf.length());

}

break;

}

}

}

}

int main() {

std::vector<std::string> urls;

urls.push\_back("https://mytona.com:325/index.html");

urls.push\_back("https://mytona.com:325/");

urls.push\_back("https://mytona.com/index.html");

urls.push\_back("https://mytona.com:8080");

urls.push\_back("https://mytona.com/index.html");

urls.push\_back("https://mytona.com:326");

urls.push\_back("https://mytona.com");

urls.push\_back("mytona.com:327");

urls.push\_back("mytona.com");

urls.push\_back("mytona.com/index.html");

urls.push\_back("mytona.com:328/index.html");

treatment(urls);

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

cout << urls.at(i) << endl;

}

return 0;

}

**Задание 4:**

Напишите программу которая открывает файл "input.txt" и удаляет в ней все пробелы и числа

**#include <iostream>**

#include <fstream>

#include <string.h>

#include <vector>

using namespace std;

void treatment(vector<string> &readfile) {

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

int j = 0;

string strbuf = readfile.at(i);

int len = readfile.at(i).length();

while (j < len) {

switch (strbuf[j]) {

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9':

case '0':

case ' ': {

char buf;

int len = strbuf.length();

for (int l = j; l < len; l++) {

buf = strbuf[l + 1];

strbuf[l] = buf;

}

strbuf = strbuf.substr(0, len - 1);

break;

}

default:

j++;

break;

}

}

readfile.at(i) = strbuf;

}

}

int main() {

setlocale(LC\_ALL, "rus");

string buff;

ifstream fin("/home/pda/CLionProjects/4task/input.txt");

vector<string> readfile;

if (!fin.is\_open()) // если файл не открыт

cout << "Файл не может быть открыт!\n"; // сообщить об этом

else {

while (!fin.eof()) {

getline(fin, buff);

readfile.push\_back(buff);

}

fin.close(); // закрываем файл

}

treatment(readfile);

ofstream fout("/home/pda/CLionProjects/4task/input.txt", ios\_base::trunc);

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

fout << readfile.at(i) << endl;

}

fout.close();

return 0;

}

**Задание 5:**

Программа ожидает ввода трёх пар целочисленных значений, представляющих собой точки в двумерном пространстве

Формат: x1 y1 x2 y2 x3 y3

Вывести на экран координаты точек кривой Безье для всех целочисленных x построенных через заданные точки. Очередность ввода начальной, конечной точек и точки опоры значения не имеют

*Все тестовые задания должны быть выполнены Вами самостоятельно и являться результатом исключительно Вашей интеллектуальной деятельности, созданы без нарушения прав третьих лиц, незаконных заимствований и плагиата. Направляя результат выполненных тестовых заданий, Вы подтверждаете, что понимаете и принимаете тот факт, что ООО "Майтона" могло получить или получать от третьих лиц, либо разработать или разрабатывать самостоятельно материалы, схожие с направленными Вами.*

*#include <iostream>*

#include <math.h>

using namespace std;

int main() {

int x1, x2, x3, y1, y2, y3;

cin >> x1;

cin >> y1;

cin >> x2;

cin >> y2;

cin >> x3;

cin >> y3;

int min, max;

min = x1;

max = x1;

if (x2 < min) {

min = x2;

if (x3 < min) { min = x3; }

}

if (x2 > max) {

max = x2;

if (x3 > max) { max = x3; }

}

float a = float(x1 - 2 \* x2 + x3);

float b = float(2 \* (x2 - x1));

float c;

float d;

for (int i = min; i <= max; i++) {

if (a == 0) {

float t = (i - x1) / (float) b;

if (t >= 0 && t <= 1) {

float y = (1 - t) \* (1 - t) \* y1 + 2 \* (1 - t) \* t \* y2 + t \* t \* y3;

cout << "x=" << i;

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

}

} else {

c = float(x1 - i);

d = (float) (b \* b - 4 \* a \* c);

if (d >= 0) {

float t1, t2;

t1 = (-b - sqrt(d)) / (float) (2 \* a);

t2 = (-b + sqrt(d)) / (float) (2 \* a);

if (t1 >= 0 && t1 <= 1) {

float y = (1 - t1) \* (1 - t1) \* y1 + 2 \* (1 - t1) \* t1 \* y2 + t1 \* t1 \* y3;

cout << "x=" << i;

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

}

if (t2 >= 0 && t2 <= 1) {

float y = (1 - t2) \* (1 - t2) \* y1 + 2 \* (1 - t2) \* t2 \* y2 + t2 \* t2 \* y3;

cout << "x=" << i;

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

}

}

}

}

return 0;

}