Екзаменаційна робота

з предмету

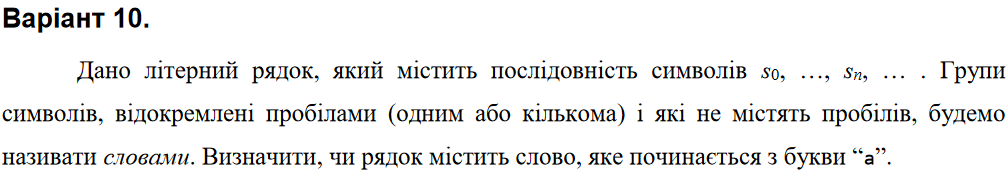
**«Алгоритмізація та програмування»**

ст. гр. ІН-105Б

Горанова Анастаса Недялковича

1. **Лабораторна №8.2**

**Умова завдання:**



**Текст програми:**

#include <string.h>

#include <iostream>

using namespace std;

bool FirstLetter(char\* str);

int main()

{

char s[101];

cout << "Enter string:" << endl;

cin.getline(s, 101);

if (FirstLetter(s))

cout << "Yes" << endl;

else

cout << "No" << endl;

}

bool FirstLetter(char\* str)

{

char\* nextToken;

char\* str2 = strtok\_s(str, " ", &nextToken);

while (str2 != NULL)

{

if (str2[0] == 'a')

return true;

str2 = strtok\_s(NULL, " ", &nextToken);

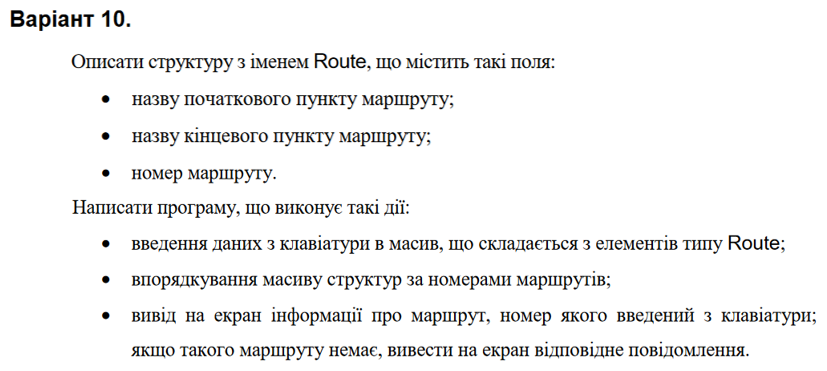
}

return false;

}

1. **Лабораторна №9.3**

**Умова завдання:**



**Текст програми:**

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

struct route

{

string start;

string end;

int nomer;

};

void Create(route\* n, const int N);

void Print(route\* n, const int N);

void Menu(route\* n, const int N);

void Sort(route\* n, const int N);

int Search(route\* n, const int N, const int nomer);

int main() {

int N;

cout << "Enter the number of routes: "; cin >> N;

route\* n = new route[N];

Create(n, N);

Print(n, N);

Menu(n, N);

}

void Create(route\* n, const int N)

{

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

{

cout << "Route #" << i + 1 << ":" << endl;

cout << " Route number: "; cin >> n[i].nomer;

cin.get();

cin.sync();

cout << " Start: "; getline(cin, n[i].start);

cout << " End: "; getline(cin, n[i].end);

cout << endl;

}

}

void Print(route\* n, const int N)

{

cout << "====================================="

<< endl;

cout << "| # | Start | End | Number " << setw(4) << "|"

<< endl;

cout << "-------------------------------------"

<< endl;

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

{

cout << "| " << setw(1) << right << i + 1 << " ";

cout << "| " << setw(7) << left << n[i].start << " "

<< "| " << setw(6) << right << n[i].end << " "

<< "| " << setw(10) << left << n[i].nomer << "|" << endl;

}

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

}

void Menu(route\* n, const int N) {

int choice;

do {

cout << "--------------------------\n";

cout << "MENU\n";

cout << "[1] - Sort\n";

cout << "[2] - Search\n";

cout << "[3] - Exit\n";

cout << "--------------------------\n";

cout << "Choose: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Sorted:\n";

Sort(n, N);

Print(n, N);

break;

case 2:

{

cout << "Search:\n";

int number, found;

cout << "Enter route number: ";

cin >> number;

found = Search(n, N, number);

if (found != -1)

{

cout << "Start: " << n[found].start << endl;

cout << "End: " << n[found].end << endl;

}

else

cout << "No route with this number was found." << endl;

}

case 3:

break;

default:

cout << "--------------------------\n";

cout << "MENU\n";

cout << "[1] - Sort\n";

cout << "[2] - Search\n";

cout << "[3] - Exit\n";

cout << "--------------------------\n";

cout << "Choose:";

cin >> choice;

}

} while (choice != 3);

}

void Sort(route\* n, const int N) {

route tmp;

for (int i0 = 0; i0 < N - 1; i0++) {

for (int i1 = 0; i1 < N - i0 - 1; i1++) {

if (n[i1].nomer > n[i1 + 1].nomer)

{

tmp = n[i1];

n[i1] = n[i1 + 1];

n[i1 + 1] = tmp;

}

}

}

}

int Search(route\* n, const int N, const int nomer) {

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

if (n[i].nomer == nomer)

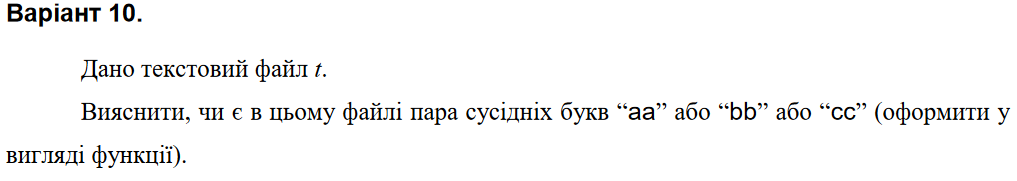
return i;

return -1;

}

1. **Лабораторна №10.1**

**Умова завдання:**



**Текст програми:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

using namespace std;

bool search(const char\* s);

int main()

{

char str[101];

ifstream f("test.txt");

f >> str;

if (search(str))

cout << "Yes" << endl;

else

cout << "No" << endl;

f.close();

return 0;

}

bool search(const char\* s)

{

for (int j = 0; s[j] != '\0'; j++)

if ((s[j] == 'a' && s[j + 1] == 'a') || (s[j] == 'b' && s[j + 1] == 'b') || (s[j] == 'c' && s[j + 1] == 'c'))

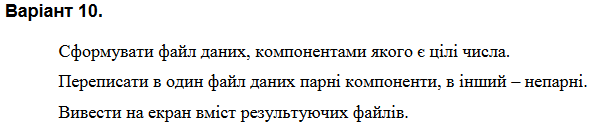
return true;

return false;

}

1. **Лабораторна №11.1**

**Умова завдання:**



**Текст програми:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

#include <stdio.h>

using namespace std;

void CreateBIN(char\* fname)

{

ofstream f(fname, ios::binary);

if (f.fail())

{

cerr << "Error opening file " << endl;

exit(1);

}

char ch;

int x;

do

{

cout << "Enter number: "; cin >> x;

f.write((char\*)&x, sizeof(x));

cout << "Continue? (y/n): "; cin >> ch;

} while (ch == 'y' || ch == 'Y');

cout << endl;

}

void PrintBIN(char\* filename)

{

ifstream f(filename, ios::binary);

if (f.fail())

{

cerr << "Error opening file " << endl;

exit(1);

}

int x;

while (f.read((char\*)&x, sizeof(x)))

cout << x << endl;

cout << endl;

}

void ProcessBIN(char\* fname, char\* outnameodd, char\* outnameeven)

{

ifstream f(fname, ios::binary);

ofstream godd(outnameodd, ios::binary);

ofstream geven(outnameeven, ios::binary);

int x;

while (f.read((char\*)&x, sizeof(x)))

{

if (x % 2 == 0)

geven.write((char\*)&x, sizeof(x));

else

godd.write((char\*)&x, sizeof(x));

}

}

int main()

{

char fname[100];

cout << "Enter input file name: "; cin >> fname;

CreateBIN(fname);

PrintBIN(fname);

char outnameodd[100]; char outnameeven[100];

cout << "Enter output file name (for odd numbers): "; cin >> outnameodd;

cout << "Enter output file name (for even numbers): "; cin >> outnameeven;

ProcessBIN(fname, outnameodd, outnameeven);

cout << "Even numbers: " << endl;

PrintBIN(outnameeven);

cout << "Odd numbers: " << endl;

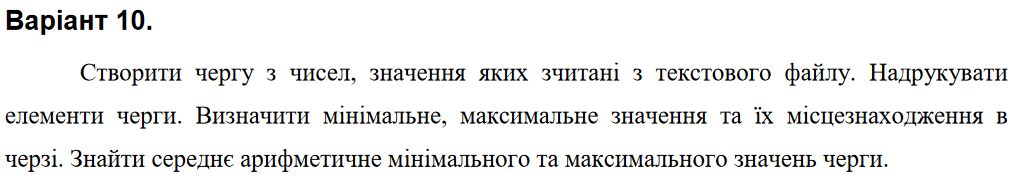
PrintBIN(outnameodd);

return 0;

}

1. **Лабораторна №12.6**

**Умова завдання:**



**Текст програми:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

typedef int Info;

struct Elem

{

Elem\* link;

int info;

};

void enqueue(Elem\*& first, Elem\*& last, Info value);

Info dequeue(Elem\*& first, Elem\*& last);

void read(Elem\*& first, Elem\*& last);

void print(Elem\* L);

void count(Elem\* L);

int main()

{

Elem\* first = NULL,

\* last = NULL;

read(first, last);

cout << endl;

print(first); // вивід черги

cout << endl;

count(first);

cout << endl;

cout << "Queue: " << endl << endl;

while (first != NULL)

cout << dequeue(first, last) << " ";

cout << endl;

return 0;

}

void enqueue(Elem\*& first, Elem\*& last, Info value)

{

Elem\* tmp = new Elem;

tmp->info = value;

tmp->link = NULL;

if (last != NULL)

last->link = tmp;

last = tmp;

if (first == NULL)

first = tmp;

}

Info dequeue(Elem\*& first, Elem\*& last)

{

Elem\* tmp = first->link;

Info value = first->info;

delete first;

first = tmp;

if (first == NULL)

last = NULL;

return value;

}

void read(Elem\*& first, Elem\*& last)

{

ifstream f("ap12.txt");

Info symbol;

while (!f.eof()) {

f >> symbol;

enqueue(first, last, symbol);

}

}

void print(Elem\* L)

{

while (L != NULL)

{

cout << L->info << " ";

L = L->link;

}

cout << endl;

}

void count(Elem\* L)

{

Info min = L->info;

Info max = L->info;

Info min\_pos = 0;

Info max\_pos = 0;

Info k = 0;

while (L->link != NULL)

{

k++;

if (L->info < min) {

min = L->info;

min\_pos = k;

}

else if (L->info > max) {

max = L->info;

max\_pos = k;

}

L = L->link;

}

if (L != NULL) {

k++;

if (L->info < min) {

min = L->info;

min\_pos = k;

}

else if (L->info > max) {

max = L->info;

max\_pos = k;

}

}

cout << "Minimum " << min << " be in position " << min\_pos << endl;

cout << "Maximum " << max << " be in position " << max\_pos << endl;

cout << "Arithmetic average " << (max + min) / 2 << endl;

}