Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: «Лабораторная работа №11»

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Желнин Н.А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

1. Постановка задачи

Реализовать односвязный и двусвязный список, организовать функции добавления элемента в начало, середину и конец, их удаление, добавление k элементов перед заданным элементом. Реализовать стэк и очередь.

1. Алгоритм решения
2. Создаем структуру узла.
3. Создаем структуру списка.
4. Создаем функцию проверки списка на наличие узлов.
5. Функции добавления и удаления элемента (в/из) начал(о/а) списка.
6. Функция добавления и удаления элемента (в/из) кон(ец/ца) списка.
7. Функция добавления и удаления элемента (в/из) середин(у/ы) списка через нахождение его длины.
8. Функция вывода списка в консоль.
9. Функция добавления k элементов.

3. Код программы

*//односвязный*

#include <iostream>

using namespace std;

struct Node {

char val;

Node\* next;

Node(char \_val) : val(\_val), next(nullptr) {}

};

struct list {

Node\* first;

Node\* last;

list() : first(nullptr), last(nullptr) {}

bool is\_empty() {

return first == nullptr;

}

void push\_front(char \_val){

Node\* p = new Node(\_val);

p->next = first;

first = p;

}

void push\_back(char \_val) {

Node\* p = new Node(\_val);

if (is\_empty()) {

first = p;

last = p;

return;

}

last->next = p;

last = p;

}

void push\_mid(char \_val){

Node\* p = new Node(\_val);

Node\* p2 = first;

int len = 0;

while (p2 != NULL) {

len++;

p2 = p2->next;

}

cout << "Dlina " << len << endl;

len = len / 2;

Node\* p3 = first;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = first;

for(int i = 0; i < len; i++) p4 = p4->next;

p3->next = p;

p->next = p4;

}

void remove\_mid(){

Node\* p2 = first;

int len = 0;

while (p2 != NULL) {

len++;

p2 = p2->next;

}

cout << "Dlina " << len << endl;

if(len % 2 == 1) len = len / 2;

else len = len / 2 -1;

Node\* p3 = first;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = first;

for(int i = 0; i < len+1; i++) p4 = p4->next;

p3->next = p4;

}

void print() {

if (is\_empty()) return;

Node\* p = first;

while (p) {

cout << p->val << " ";

p = p->next;

}

cout << endl;

}

Node\* find(char \_val) {

Node\* p = first;

while (p && p->val != \_val) p = p->next;

return (p && p->val == \_val) ? p : nullptr;

}

void remove\_first() {

if (is\_empty()) return;

Node\* p = first;

first = p->next;

delete p;

}

void remove\_last() {

if (is\_empty()) return;

if (first == last) {

remove\_first();

return;

}

Node\* p = first;

while (p->next != last) p = p->next;

p->next = nullptr;

delete last;

last = p;

}

void push\_k(char \_val, int numb){

Node\* p = new Node(\_val);

Node\* p2 = first;

Node\* p3 = first;

for(int i = 0; i<numb-2 ; i++){

p2 = p2->next;

}

for(int i = 0; i<numb-1 ; i++){

p3 = p3->next;

}

p2->next = p;

p->next = p3;

}

void delete\_k(int numb){

Node\* p = first;

Node\* p2 = first;

for(int i = 0; i<numb-3 ; i++){

p = p->next;

}

for(int i = 0; i<numb-1 ; i++){

p2 = p2->next;

}

p->next = p2;

}

};

int main()

{

char a, b;

int k, obj, obj2, k2;

int numb\_nodes;

list l;

cout << "Write a number of values: " << endl;

cin >> numb\_nodes;

while(numb\_nodes>0){

cin >> a;

l.push\_back(a);

numb\_nodes--;

}

cout << "Values: ";

l.print();

l.push\_back('r');

l.print();

l.push\_front('f');

l.print();

l.push\_mid('u');

l.print();

l.remove\_first();

l.print();

l.remove\_last();

l.print();

l.remove\_mid();

l.print();

cout << "Write a k: " << endl;

cin >> k;

cout << "Write a numb before push: " << endl;

cin >> obj;

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

cin >> b;

l.push\_k(b, obj);

obj++;

}

l.print();

cout << "Write a k2: " << endl;

cin >> k2;

cout << "Write a numb before delete: " << endl;

cin >> obj2;

for(int i = 0; i<k2; i++){

l.delete\_k(obj2);

obj2--;

}

l.print();

return 0;

}

*//двусвязный*

#include <iostream>

using namespace std;

struct Node {

char val;

Node\* next;

Node\* prev;

Node(char \_val) : prev(nullptr), val(\_val), next(nullptr) {}

};

struct list {

Node\* first;

Node\* last;

list() : first(nullptr), last(nullptr) {}

bool is\_empty() {

return first == nullptr;

}

void push\_front(char \_val){

Node\* p = new Node(\_val);

p->next = first;

first = p;

}

void push\_back(char \_val) {

Node\* p = new Node(\_val);

if (is\_empty()) {

first = p;

last = p;

return;

}

last->next = p;

last = p;

}

void push\_mid(char \_val){

Node\* p = new Node(\_val);

Node\* p2 = first;

int len = 0;

while (p2 != NULL) {

len++;

p2 = p2->next;

}

cout << "Dlina " << len << endl;

len = len / 2;

Node\* p3 = first;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = first;

for(int i = 0; i < len; i++) p4 = p4->next;

p3->next = p;

p->prev = p3;

p->next = p4;

p4->prev = p;

}

void remove\_mid(){

Node\* p2 = first;

int len = 0;

while (p2 != NULL) {

len++;

p2 = p2->next;

}

cout << "Dlina " << len << endl;

if(len % 2 == 1) len = len / 2;

else len = len / 2 -1;

Node\* p3 = first;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = first;

for(int i = 0; i < len+1; i++) p4 = p4->next;

p3->next = p4;

p4->prev = p3;

}

void print() {

if (is\_empty()) return;

Node\* p = first;

while (p) {

cout << p->val << " ";

p = p->next;

}

cout << endl;

}

Node\* find(char \_val) {

Node\* p = first;

while (p && p->val != \_val) p = p->next;

return (p && p->val == \_val) ? p : nullptr;

}

void remove\_first() {

if (is\_empty()) return;

Node\* p = first;

first = p->next;

first->prev = nullptr;

delete p;

}

void remove\_last() {

if (is\_empty()) return;

if (first == last) {

remove\_first();

return;

}

Node\* p = first;

while (p->next != last) p = p->next;

p->next = nullptr;

delete last;

last = p;

}

void push\_k(char \_val, int numb){

Node\* p = new Node(\_val);

Node\* p2 = first;

Node\* p3 = first;

if(numb == 1){

p->next = first;

first->prev = p;

first = p;

return;

}

for(int i = 0; i<numb-2 ; i++){

p2 = p2->next;

}

for(int i = 0; i<numb-1 ; i++){

p3 = p3->next;

}

p2->next = p;

p->prev = p2;

p->next = p3;

p3->prev = p;

}

void delete\_k(int numb){

Node\* p = first;

Node\* p2 = first;

for(int i = 0; i<numb-3 ; i++){

p = p->next;

}

for(int i = 0; i<numb-1 ; i++){

p2 = p2->next;

}

p->next = p2;

p2->prev = p;

}

};

int main()

{

char a, b;

int k, obj, obj2, k2;

int numb\_nodes;

list l;

cout << "Write a number of values: " << endl;

cin >> numb\_nodes;

while(numb\_nodes>0){

cin >> a;

l.push\_back(a);

numb\_nodes--;

}

cout << "Values: ";

l.print();

l.push\_back('r');

l.print();

l.push\_front('f');

l.print();

l.push\_mid('u');

l.print();

l.remove\_first();

l.print();

l.remove\_last();

l.print();

l.remove\_mid();

l.print();

cout << "Write a k: " << endl;

cin >> k;

cout << "Write a numb before push: " << endl;

cin >> obj;

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

cin >> b;

l.push\_k(b, obj);

obj++;

}

l.print();

cout << "Write a k2: " << endl;

cin >> k2;

cout << "Write a numb before delete: " << endl;

cin >> obj2;

for(int i = 0; i<k2; i++){

l.delete\_k(obj2);

obj2--;

}

l.print();

return 0;

}

//стэк

#include <iostream>

using namespace std;

struct Node {

char val;

Node\* next;

Node(char \_val) : val(\_val), next(nullptr) {}

};

struct list {

Node\* top;

list() : top(nullptr) {}

bool is\_empty() {

return top == nullptr;

}

void push\_top(char \_val){

Node\* p = new Node(\_val);

p->next = top;

top = p;

}

void push\_down(char \_val) {

Node\* p = new Node(\_val);

Node\* temp = top;

while (temp->next != NULL) {

temp = temp->next;

}

if (is\_empty()) {

top = p;

return;

}

temp->next = p;

}

void push\_mid(char \_val){

Node\* p = new Node(\_val);

Node\* temp = top;

int len = 0;

while (temp != NULL) {

len++;

temp = temp->next;

}

//cout << "Dlina " << len << endl;

len = len / 2;

Node\* p3 = top;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = top;

for(int i = 0; i < len; i++) p4 = p4->next;

p3->next = p;

p->next = p4;

}

void remove\_mid(){

Node\* p2 = top;

int len = 0;

while (p2 != NULL) {

len++;

p2 = p2->next;

}

//cout << "Dlina " << len << endl;

if(len % 2 == 1) len = len / 2;

else len = len / 2 -1;

Node\* p3 = top;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = top;

for(int i = 0; i < len+1; i++) p4 = p4->next;

p3->next = p4;

}

void print() {

if (is\_empty()) return;

Node\* p = top;

while (p) {

cout << p->val << endl;

p = p->next;

}

cout << endl;

}

void remove\_top() {

if (is\_empty()) return;

Node\* p = top;

top = p->next;

delete p;

}

void remove\_down() {

int k = 0;

if (is\_empty()) return;

if (top->next == NULL) {

remove\_top();

return;

}

Node\* p = top;

while (p->next != NULL){

p = p->next;

k++;

}

Node\* p2 = top;

for(int i = 0; i<k-1;i++) p2=p2->next;

p2->next = nullptr;

delete p;

}

};

void print\_menu(){

cout << "1. Push top" << endl;

cout << "2. Push down" << endl;

cout << "3. Push mid" << endl;

cout << "4. Remove top" << endl;

cout << "5. Remove down" << endl;

cout << "6. Remove mid" << endl;

cout << "7. Print stack" << endl;

cout << "8. End" << endl;

}

list l;

void menu(){

int x;

bool f = true;

char a;

while(f){ print\_menu(); cin >> x; cout << endl;

switch(x){

case 1:

cout << "Write an letter: ";

cin >> a;

cout << endl;

l.push\_top(a);

break;

case 2:

cout << "Write an letter: ";

cin >> a;

cout << endl;

l.push\_down(a);

break;

case 3:

cout << "Write an letter: ";

cin >> a;

cout << endl;

l.push\_mid(a);

break;

case 4:

l.remove\_top();

break;

case 5:

l.remove\_down();

break;

case 6:

l.remove\_mid();

break;

case 7:

l.print();

cout << endl;

break;

case 8:

f = false;

break;

default:

cout << "No, man" << endl;

break;

}

}

}

int main()

{

menu();

return 0;

}

//очередь

#include <iostream>

using namespace std;

struct Node {

char val;

Node\* next;

Node(char \_val) : val(\_val), next(nullptr) {}

};

struct list {

Node\* first;

Node\* last;

list() : first(nullptr), last(nullptr) {}

bool is\_empty() {

return first == nullptr;

}

void add\_first(char \_val){

Node\* p = new Node(\_val);

if (is\_empty()) {

first = p;

last = p;

return;

}

first->next = p;

first = p;

}

void add\_last(char \_val) {

Node\* p = new Node(\_val);

if (is\_empty()) {

first = p;

last = p;

return;

}

p->next = last;

last = p;

}

void add\_mid(char \_val){

Node\* p = new Node(\_val);

Node\* temp = last;

int len = 0;

while (temp != NULL) {

len++;

temp = temp->next;

}

len = len / 2;

Node\* p3 = last;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = last;

for(int i = 0; i < len; i++) p4 = p4->next;

p3->next = p;

p->next = p4;

}

void remove\_mid(){

Node\* p2 = last;

int len = 0;

while (p2 != NULL) {

len++;

p2 = p2->next;

}

if(len % 2 == 1) len = len / 2;

else len = len / 2 -1;

Node\* p3 = last;

for(int i = 0; i < len-1; i++) p3 = p3->next;

Node\* p4 = last;

for(int i = 0; i < len+1; i++) p4 = p4->next;

p3->next = p4;

}

void print() {

if (is\_empty()) return;

Node\* p = last;

while (p) {

cout << p->val << endl;

p = p->next;

}

cout << endl;

}

void remove\_first() {

Node\* p = last;

if (is\_empty()) return;

int k = 0;

if (last->next == NULL) {

remove\_last();

return;

}

while (p->next != NULL){

p = p->next;

k++;

}

Node\* p2 = last;

for(int i = 0; i<k-1;i++) p2=p2->next;

p2->next = nullptr;

first = p2;

delete p;

}

void remove\_last() {

Node\* p = last;

last = p->next;

delete p;

}

};

void print\_menu(){

cout << "1. Add first" << endl;

cout << "2. Add last" << endl;

cout << "3. Add mid" << endl;

cout << "4. Remove first" << endl;

cout << "5. Remove last" << endl;

cout << "6. Remove mid" << endl;

cout << "7. Print ochered" << endl;

cout << "8. End" << endl;

}

list l;

void menu(){

int x;

bool f = true;

char a;

while(f){ print\_menu(); cin >> x; cout << endl;

switch(x){

case 1:

cout << "Write an letter: ";

cin >> a;

cout << endl;

l.add\_first(a);

break;

case 2:

cout << "Write an letter: ";

cin >> a;

cout << endl;

l.add\_last(a);

break;

case 3:

cout << "Write an letter: ";

cin >> a;

cout << endl;

l.add\_mid(a);

break;

case 4:

l.remove\_first();

break;

case 5:

l.remove\_last();

break;

case 6:

l.remove\_mid();

break;

case 7:

l.print();

cout << endl;

break;

case 8:

f = false;

break;

default:

cout << "No, man" << endl;

break;

}

}

}

int main()

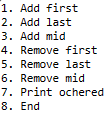
{

menu();

return 0;

}

4. Вывод в консоль

Write a number of values:

4

a b c d

Values: a b c d

a b c d r

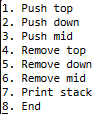
f a b c d r

Dlina 6

f a b u c d r

a b u c d r

a b u c d

Dlina 5

a b c d

Write a k:

3

Write a numb before push:

2

y u i

a y u i b c d