

Вариант 3

1) Задание 1. Выполнить упражнения и представить процесс выполнения в отчете

1. Провести преобразование инфиксной записи выражения в префиксную нотацию, расписывая процесс по шагам $S = a + (b - c * k) - d * e - f$ Решение:

$$c * k = * c k b - c *$$

$$k = - b * c k d * e$$

$$= * d e$$

$$a + (b - c * k) = + a - b * c k$$

$$a + (b - c * k) - d * e - f = - + a - b * c k * d e f$$
 Ответ:

$$- + a - b * c k * d e f$$

2. Представить постфиксную нотацию выражений

$$a + (c - b) / (b * d)$$

Решение:

$$c - b = c b - b$$

$$* d = b d *$$

$$(c - b) / (b * d) = c b - b d * / a + (c$$

$$- b) / (b * d) = a c b - b d * / +$$

Ответ: $a c b - b d * / +$

$$(a + b) * c - (d + e * f / ((g / h + i - j) * k)) / r$$

Решение:

$$a + b = a b +$$

$$(a + b) * c = a b + c * g$$

$$/ h = g h / g / h + i = g h$$

$$/ i + g / h + i - j = g h / i$$

$$+ j -$$

$$(g / h + i - j) * k = g h / i + j - k * e$$

$$* f = e f *$$

$$e * f / ((g / h + i - j) * k) = e f * g h / i + j - k * / d + e *$$

$$f / ((g / h + i - j) * k) = d e f * g h / i + j - k * / +$$

$$(d + e * f / ((g / h + i - j) * k)) / r = d e f * g h / i + j - k * / + r /$$

$$(a + b) * c - (d + e * f / ((g / h + i - j) * k)) / r = a b + c * d e f * g h / i + j - k * / + r / -$$

Ответ: $a b + c * d e f * g h / i + j - k * / + r / -$

3. Представить префиксную нотацию выражений п.2

$$a + (c - b) / (b * d)$$

Решение:

$$c - b = - c b b$$

$$* d = * b d$$

$$(c - b) / (b * d) = / - c b * b d a + (c$$

$$- b) / (b * d) = + a / - c b * b d$$

Ответ: $+ a / - c b * b d$

$$(a + b) * c - (d + e * f / ((g / h + i - j) * k)) / r a$$

$$+ b = + a b$$

$$(a + b) * c = * c + a b$$

$$g / h = / g h \quad g / h + i =$$

$$+ / g h i$$

$$g / h + i - j = - + / g h i j$$

$$(g / h + i - j) * k = * - + / g h i j k$$

$$e * f = * e f$$

$$e * f / ((g / h + i - j) * k) = / * e f * - + / g h i j k d + e$$

$$* f / ((g / h + i - j) * k = + d / * e f * - + / g h i j k$$

$$(d + e * f / ((g / h + i - j) * k)) / r = / + d / * e f * - + / g h i j k r$$

$$(a + b) * c - (d + e * f / ((g / h + i - j) * k)) / r = - * c + a b / + d / * e f * - + / g h i j k r$$

Ответ: $- * c + a b / + d / * e f * - + / g h i j k r$

4. Провести вычисление значения выражения представленного в постфиксной форме, расписывая процесс по шагам $7 2 - 3 2 + *$ Решение:

$$7 2 - = 7 - 2 = 5$$

$$2 3 + = 2 + 3 = 5$$

$$7 2 - 3 2 + * = 5 * 5 = 25$$

2) Задание 2. Выполнить программную реализацию следующих задач

1. Реализовать операции стек: втолкнуть элемент в стек, вытолкнуть элемент из стека, вернуть значение элемента в вершине стека, сделать стек пустым, определить, пуст ли стек. Рассмотреть два варианта реализации: на массиве (или строке); на однонаправленном списке.

- Создать класс или просто заголовочный файл с функциями.

Решение:

Реализация на однонаправленном списке.

Файл main.cpp

```
#include <iostream>
#include <windows.h>
#include "stack.h" using
namespace std;

void wod(char fail, int& d)
{
    while (cin.fail())
    {
        char fail;
cin.clear();
cin >> fail;
        cout << "Ошибка. Введите значение заново\n\n";
        cin >> d;
    }
}

int main() {
    setlocale(LC_ALL, "Rus");
```

```

        SetConsoleCP(1251);
SetConsoleOutputCP(1251);
        stack st;
        int n;
        cout << "Введите n ";
cin >> n;
        if (cin.fail()) wod(cin.fail(), n);
        cout << "\nПроверим пуст ли список...\n";
st.trueempty();        cout << "\nСоздание
списка\n";        st.createList(n);        st.outList();
st.push(n);        st.outList();        st.pop(n);
st.outList();        st.head();
        cout << "\nОчистим список и проверим пуст ли он...\n";
st.empty();        st.trueempty();
}

```

Файл stack.h

```

#include <iostream>
#include <list>
using namespace std;
#ifndef STACK_H
#define STACH_H
class
stack {
public:
        typedef int Linfo;
        struct Lnode
        {
                Linfo info;
                Lnode* next;
        };
        Lnode* HeadList;
Lnode* L = NULL;        void
wodn(char, int&);        void
createlist(int);        void
outList();        void push(int);
void pop(int);        void
empty();        void head();
        void trueempty();
};
#endif

```

Файл stack.cpp

```

#include "stack.h"

void stack::trueempty()
{ if (L == NULL) cout << "Список пуст\n"; else
cout << "Список не пуст\n";
}

void stack::wodn(char fail, int& d)
{
        while (cin.fail())
        {
                char fail;
cin.clear();
cin >> fail;
                cout << "Ошибка. Введите значение заново\n\n";
}
}

```

```

        cin >> d;
    }
}

void stack::createList(int n)
{
    Lnode* q = NULL, * q1 = NULL; cout <<
    "Введите " << n << " чисел\n";
    for (int i = 1; i <= n; i++)
    {
        if (q == NULL)
        {
            q = new Lnode;
            cin >> q->info;
            if (cin.fail()) wodn(cin.fail(), q->info);
            q->next = NULL;
        }
        else
        {
            q1 = new Lnode;
            cin >> q1->info;
            if (cin.fail()) wodn(cin.fail(), q1->info);
            q1->next = q;
        }
        q = q1;
    }
    L = q1;
}

void stack::outList()
{
    Lnode* node = L; cout <<
    "Список:\n"; while (node
    != 0)
    {
        cout << node->info << " ";
        node = node->next;
    }
    cout << endl;
}

void stack::push(int n)
{
    int b;
    cout << "Добавим элемент...\n";
    cout << "Значение элемента = "; cin >>
    b;
    Lnode* q = NULL, * q1 = NULL, * q4 = NULL;
    for (int i = 1; i <= n + 1; i++)
    {
        q = new Lnode;
        if (i != 1)
        {
            q->info = L->info;
        }
        else q->info = b; q->next =
        NULL; if (q4 == NULL)
        {
            q4 = q;
            q1 = q4;
        }
    }
}

```

```

        else
        {
            q1->next = q;
            q1 = q;
        }
        if (i != 1) L = L->next;
    }
    L = q4;
}

void stack::pop(int n)
{
    cout << "Удалим элемент...\n"; Lnode* q =
    NULL, * q1 = NULL, * q4 = NULL;
    for (int i = 1; i <= n + 1; i++)
    {
        if (i!= 1)
        {
            q = new Lnode;
            q->info = L->info;
            q->next = NULL;
            if (q4 == NULL)
            {
                q4 = q;
                q1 = q4;
            }
            else
            {
                q1->next = q;
                q1 = q;
            }
        }
        L = L->next;
    }
    L = q4;
}

void stack::head()
{
    cout << "Значение элемента в вершине стека = " << L->info << endl;
}

void stack::empty()
{
    L = NULL;
}

```

Тестирование. Создание списка и вывод на экран списка.

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
1	n = 5 1 2 3 4 5	5 4 3 2 1	5 4 3 2 1
2	n = 10 1 2 3 4 5 6 7 8 9 10	10 9 8 7 6 5 4 3 2 1	10 9 8 7 6 5 4 3 2 1
3	n = 1 4	4	4

4	n = 2 1 3	3 1	3 1
5	n = 3 8 5 3	3 5 8	3 5 8

Тестирование. Вставка элемента и вывод на экран списка.

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
		данные	программы
1	Значение: 1 5 4 3 2 1	1 5 4 3 2 1	1 5 4 3 2 1
2	Значение: 5 5 4 3 2 1	5 5 4 3 2 1	5 5 4 3 2 1
3	Значение: 0 2 1	0 2 1	0 2 1
4	Значение: 4 3 3 3	4 3 3 3	4 3 3 3
5	Значение: 10 1 2 3 4 5	10 1 2 3 4 5	10 1 2 3 4 5

Тестирование. Удаление элемента и вывод на экран списка.

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
1	1 2 3 4 5	2 3 4 5	2 3 4 5
2	2 3 4 5	3 4 5	3 4 5
3	3 4 5	4 5	4 5
4	1 1 1	1 1	1 1
5	1 0 9 4 4	0 9 4 4	0 9 4 4

Реализация на массиве.

main.cpp

```
#include <iostream> using
namespace std;
#include "stack.h"
#include <windows.h>

void wod(char fail, int& d)
{
    while (cin.fail())
    {
        char fail;
cin.clear();
cin >> fail;
        cout << "Ошибка. Введите значение заново\n\n";
        cin >> d;
    }
}
int
main() {
    setlocale(LC_ALL, "Rus");
```

```

        SetConsoleCP(1251);
SetConsoleOutputCP(1251);
        int n;
        cout << "Введите n ";
        cin >> n;
        if (cin.fail()) wout(cin.fail(), n);
cout << "\nПроверим пуст ли стек...\n";
        st stk;
        stk.trueempty();
cout << "\nСоздание\n";
stk.createst(n);
stk.outst(n);
stk.push(n);
stk.outst(n);
stk.pop(n);  stk.outst(n);
        stk.head();
        cout << "\nОчистим стек и проверим пуст ли он...\n";
stk.empty();          stk.trueempty();
        return 0;
}

```

stack.h

```

#ifndef ST_H
#define ST_H #include
<iostream> using
namespace std;
class
st
{
public:
int s;
        int *massiw = NULL;
int front;  void
trueempty();      void
createst(int);    void
outst(int);  void
push(int &);      void
pop(int &);  void head();
        void empty();
};
#endif

```

stack.cpp

```

#include "stack.h"

void st::trueempty()
{
    if (massiw == NULL) cout << "Стек пуст"; else
cout << "Стек не пуст";
}

void st::createst(int s)
{
    massiw = new int [s];
    for (int i = 0; i < s; i++)
    {
        cin >> massiw[s-i-1];
        while (cin.fail())

```

```

        {
            char fail;
            cin.clear();
cin >> fail;
            cout << "Ошибка. Введите значение заново\n\n";
            cin >> massiw[i];
        }
    }
}

void st::outst(int n)
{
    for (int i = 0; i < n; i++)
        cout << massiw[i] << " ";
    cout << "\n";
}

void st::empty()
{
    massiw = NULL;
}

void st::push(int &n)
{
    int j, k;
    cout << "Введите значение\n";
cin >> j;    n++;
    k = n - 1;
    massiw = (int*)realloc(massiw, n * sizeof(int));
    while (k > 0)
    {
        massiw[k] = massiw[k - 1];
        k--;
    }
    massiw[0] = j;
}

void st::pop(int& n)
{
    n--;
    int i = 0;
    while (i < n)
    {
        massiw[i] = massiw[i + 1];
        i++;
    }
    if (n != 0)
    {
        massiw = (int*)realloc(massiw, n * sizeof(int));
    }
    else {
        massiw = NULL;
    }
}

void st::head() {
    cout << massiw[0] << endl;
}

```

Тестирование. Создание списка и вывод на экран списка.

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
1	n = 5 1 2 3 4 5	5 4 3 2 1	5 4 3 2 1
2	n = 10 1 2 3 4 5 6 7 8 9 10	10 9 8 7 6 5 4 3 2 1	10 9 8 7 6 5 4 3 2 1
3	n = 1 4	4	4
4	n = 2 1 3	3 1	3 1
5	n = 3 8 5 3	3 5 8	3 5 8

Тестирование. Вставка элемента и вывод на экран списка.

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
1	Значение: 1 5 4 3 2 1	1 5 4 3 2 1	1 5 4 3 2 1
2	Значение: 5 5 4 3 2 1	5 5 4 3 2 1	5 5 4 3 2 1
3	Значение: 0 2 1	0 2 1	0 2 1
4	Значение: 4 3 3 3	4 3 3 3	4 3 3 3
5	Значение: 10 1 2 3 4 5	10 1 2 3 4 5	10 1 2 3 4 5

Тестирование. Удаление элемента и вывод на экран списка.

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
1	1 2 3 4 5	2 3 4 5	2 3 4 5
2	2 3 4 5	3 4 5	3 4 5
3	3 4 5	4 5	4 5
4	1 1 1	1 1	1 1
5	1 0 9 4 4	0 9 4 4	0 9 4 4

2. Разработать функцию(ИИ) вычисления значения выражения, представленного в префиксной форме.

```
#include <iostream>
#include <string>
#include <functional>
#include <vector>
#include <map>
#include <windows.h>
using namespace std;

int calculate(string expression)
```

```

{
    map<char, function<int(const int&, const int&>> op;
op['+'] = [](const int& a, const int& b) { return a + b; };
op['-'] = [](const int& a, const int& b) { return a - b; };
op['*'] = [](const int& a, const int& b) { return a * b; };
op['/'] = [](const int& a, const int& b) { return a / b; };
vector<int> stack_;    int num = 0;    bool f = true;
    for (const int& i : expression)
    {
        if (isdigit(i))
        {
            num *=
10;            num += (i -
'0');            f = true;
        }
        else
        {
            if (i != ' ')
            {
                int num2 = stack_.back();
stack_.pop_back();            int
num1 = stack_.back();
stack_.pop_back();
                stack_.push_back(op[i](num1, num2));
f = false;
            }
            else if (i == ' ' && f)
            {
                stack_.push_back(num);
num = 0;
            }
        }
    }
    return stack_.back();
}

int main() {
setlocale(LC_ALL, "Rus");
    SetConsoleCP(1251);
SetConsoleOutputCP(1251);
string expression;    cout <<
"Введите выражение ";
    getline(cin, expression);
    cout << "Ответ: " << calculate(expression);
return 0;
}

```

Тестирование:

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
1	8 9 + 1 7 - *	-102	-102
2	7 2 - 3 2 + *	25	25
3	8 8 *	64	64
4	7 2 3 * 5 8 2 / - + 1 - +	13	13
5	100 0 *	0	0

3. Разработать программу сложения двух больших целых чисел (не попадающих в диапазон стандартных типов), вводимых с клавиатуры, как последовательность символов.

```
#include <iostream>;
```

```

#include <windows.h>;
#include <string>; using
namespace std;

int main() {
    setlocale(LC_ALL, "Rus");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251); string
a, b, sum; int i, j, k, v = 0;
bool f = true; cout <<
"Введите 1 число "; cin >>
a;
    cout << "Введите 2 число ";
cin >> b;
    if (a.length() > b.length())
    {
        i = a.length() - 1;
        j = b.length() - 1;
    }
    else
    {
        i = b.length() - 1;
        j = a.length() - 1;
f = false;
    }
    for (i; i >= 0; i--)
    {
        if (i >= 0 and j >= 0)
        {
            if (f)
            {
                k = (int(a[i]) - 48) + (int(b[j]) - 48);
            }
            else k = (int(b[i]) - 48) + (int(a[j]) - 48);
            v = v + k % 10;
        }
        else if (f) {
            v = v + (int(a[i]) - 48);
        }
        else if (!f) {
v = v + (int(b[i]) - 48);
        }
        k = k / 10;
        if (v >= 10)
        {
            k = k + 1;
            v = v % 10;
        }
        sum = (char)((((int)'0') + v) + sum;
        v = k;
        if (i == 0 and k != 0) sum = (char)((((int)'0') + k) + sum;
        j = j - 1;
    }
    cout << "Сумма чисел = " << sum;
return(0);
}

```

Тестирование:

№	Введенные данные	Ожидаемые выходные данные	Выходные данные программы
---	------------------	---------------------------	---------------------------

1	Первое число: 11111111111111111111 11111111 Второе число: 99999999999999999999 99999999	111111111111111111111111 1111110	111111111111111111111111 1111110
2	Первое число: 666555 Второе число: 111999	778554	778554
3	Первое число: 0 Второе число: 1111111111111111111111 1111111111111111111111 1111111111111111	111111111111111111111111 1111111111111111111111 111111111111	111111111111111111111111 1111111111111111111111 111111111111
4	Первое число: 99999999999999999999 99999999999999999999 99999999999999999999 99999999999999999999 99 Второе число: 1	1000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000	1000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000
5	Первое число: 1 Второе число: 99999999999999999999 99999999999999999999 99999999999999999999 99999999999999999999 99	1000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000	1000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000 0000000000000000000000