



ФАКУЛТЕТ ЗА ЕЛЕКТРОТЕХНИКА И ИНФОРМАЦИСКИ ТЕХНОЛОГИИ

СТРУКТУРИ СО ПРОГРАМИРАЊЕ 2017/2018

АУДИТОРИСКИ ВЕЖБИ 4
МАГАЦИНИ (ВО С И С++)

Податочни структури и приграмирање

ЗАДАЧИ:

1. Да се напише функција која како аргумент добива декаден број и променлива од тип магацин. Функцијата треба да го испечати бинарниот број, но со користење само на магацинот. Да се напишат сите структури и функции кои се потребни за да се реализира работата на оваа функција. Во главната програма да се провери работата на функцијата.

Решение 1 (во C):

```
#include <stdio.h>
#include <stdbool.h>

#define MAX 20

typedef struct element
{
    int info;
} element;

typedef struct magacin
{
    element niza[MAX];
    int vrv;
} magacin;

void inicijaliziraj(magacin *m)
{
    m->vrv = -1;
}

bool ePrazen(magacin *m)
{
    return(m->vrv == -1);
}

bool ePoln(magacin *m)
{
    return (m->vrv == MAX - 1);
}

void vmetni(magacin *m, int e)
{
    if (ePoln(m))
    {
        printf("Magacinot e poln i ne mozhe da se vmetnuvaat elementi!\n");
        fflush(stdout);
        return;
    }
    m->niza[++m->vrv].info = e;
}

element izvadi(magacin *m)
{
    if (ePrazen(m))
    {
        printf("Magacinot e prazen i ne mozhe da se vadat elementi!\n");
        fflush(stdout);
        return;
    }
    return m->niza[m->vrv--];
}
```

```
void DekVoBin(int dek, magacin *m)
{
    int cif = dek;
    inicijaliziraj(m); /*pred da vmetnuvame elementi vo magacinot, mora da go
inicijalizirame!!!*/
    while (dek != 0) //go zemame ostatokot i go smestuvame vo magacin
    {
        vmetni(m, dek % 2);
        dek /= 2;
    }
    //kako gi vadime elementite od magacinot, taka i gi pechatime
    //so shto go dobivame binarniot ekvivalent na dek
    printf("Binarniot ekvivalent na %d e ", cif); fflush(stdout);
    while (!ePrazen(m))
    {
        cif = izvadi(m).info;
        printf("%d", cif); fflush(stdout);
    }
    printf("\n"); fflush(stdout);
}

int main()
{
    magacin mag;
    int dek;
    printf("Vnesete dekadni broj: "); fflush(stdout);
    scanf("%d", &dek);
    DekVoBin(dek, &mag);
}
```

Решение 2 (во C++, функции во структура):

```
#include <iostream>
using namespace std;

int const MAX = 20;

struct element
{
    int info;
};

struct magacin
{
    element niza[MAX];
    int vrv;
    void inicijaliziraj();
    bool ePrazen();
    bool ePoln();
    void vmetni(int e);
    element izvadi();
};

void magacin::inicijaliziraj()
{
    vrv = -1;
}

bool magacin::ePrazen()
{
    return(vrv == -1);
}
```

```
}

bool magacin::ePoln()
{
    return (vrv == MAX - 1);
}

void magacin::vmetni(int e)
{
    if (ePoln())
    {
        cout << "Magacinot e poln i ne mozhe da se vmetnuvaat elementi!" << endl;
        return;
    }

    niza[++vrv].info = e;
}

element magacin::izvadi()
{
    if (ePrazen())
    {
        cout << "Magacinot e prazen i ne mozhe da se vadat elementi!" << endl;
        return{};
    }
    return niza[vrv--];
}

void DekVoBin(int dek, magacin &m)
{
    int cif = dek;
    m.inicijaliziraj(); //pred da vmetnuvame elementi vo magacinot, mora da go
    inicijalizirame!!!
    while (dek != 0) //go zemame ostatokot i go smestuvame vo magacin
    {
        m.vmetni(dek % 2);
        dek /= 2;
    }
    //kako gi vadime elementite od magacinot, taka i gi pechatime
    //so shto go dobivame binarniot ekvivalent na dek
    cout << "Binarniot ekvivalent na " << cif << " e ";
    while (!(m.ePrazen()))
    {
        cif = m.izvadi().info;
        cout << cif;
    }
    cout << endl;
}

int main()
{
    magacin mag;
    int dek;
    cout << "Vnesete dekadnen broj: ";
    cin >> dek;
    DekVoBin(dek, mag);
}
```

2. Да се провери дали во даден аритметички израз заградите се балансираани.

Пример: Балансирани загради {1+[2-5]+(2+{6*3})/1}
 Небалансираани загради {1+[2-5}+(2+{6*3})/1}

Решение 1 (во C):

```
#include <stdio.h>
#include <stdbool.h>

#define MAX 20

typedef struct element
{
    char info; /* за магацинот да работи со знаци, единствено го менуваме овој дел!!!
*/
} element;

typedef struct stack
{
    element array[MAX];
    int top;
} stack;

void init(stack *s)
{
    s->top = -1;
}

bool isEmpty(stack *s)
{
    return(s->top == -1);
}

bool isFull(stack *s)
{
    return (s->top == MAX - 1);
}

void push(stack *s, element e)
{
    if (isFull(s))
    {
        printf("Magacinot e poln i ne mozhe da se vmetnuvaat elementi\n");
        fflush(stdout);
        return;
    }

    s->array[++s->top] = e;
}

element pop(stack *s)
{
    if (isEmpty(s))
    {
        printf("Magacinot e prazen i ne mozhe da se vadat elementi!\n");
        fflush(stdout);
        return;
    }

    return s->array[s->top--];
}
```

```
}

element peek(stack *s)
{
    if (isEmpty(s))
    {
        printf("Magacinot e prazen i ne mozhe da se vadat elementi!");
        fflush(stdout);
        return;
    }
    return s->array[s->top];
}

bool proverizagradi(char *izraz, stack *magacin)
{
    element e;
    init(magacin);
    for (int i = 0; i < strlen(izraz); i++) {
        if (izraz[i] == '{' || izraz[i] == '[' || izraz[i] == '(') {
            e.info = izraz[i];
            push(magacin, e);
        }
        else if (izraz[i] == '}')
            if (peek(magacin).info == '{')
                pop(magacin);
            else return false;
        else if (izraz[i] == ']')
            if (peek(magacin).info == '[')
                pop(magacin);
            else
                return false;
        else if (izraz[i] == ')')
            if (peek(magacin).info == '(')
                pop(magacin);
            else
                return false;
    }
    if (isEmpty(magacin))
        return true;
    else
        return false;
}

int main()
{
    stack magacin;
    char izraz[100];
    printf("Vnesete aritmetichki izraz: ");
    fflush(stdout);
    scanf("%s", izraz);
    if (proverizagradi(izraz, &magacin))
    {
        printf("Izrazot %s ima balansirani zagradi", izraz); fflush(stdout);
    }
    else
    {
        printf("Izrazot %s nema balansirani zagradi", izraz); fflush(stdout);
    }
}
```

Решение 2 (во C++):

```
#include <iostream>
#include <string>
using namespace std;
int const MAX = 20;

struct element
{
    char info; /* за магацинот да работи со знаци, единствено го менуваме овој дел!!!
*/
};

struct Stack
{
    element array[MAX];
    int top;
    void init();
    bool isEmpty();
    bool isFull();
    void push(element e);
    element pop();
    element peek();
};

void Stack::init()
{
    top = -1;
}

bool Stack::isEmpty()
{
    return(top == -1);
}

bool Stack::isFull()
{
    return (top == MAX - 1);
}

void Stack::push(element e)
{
    if (isFull())
    {
        cout << "Magacinot e poln i ne mozhe da se vmetnuvaat elementi!" << endl;
        return;
    }

    array[++top] = e;
}

element Stack::pop()
{
    if (isEmpty())
    {
        cout << "Magacinot e prazen i ne mozhe da se vadat elementi!" << endl;
        return{};
    }
    return array[top--];
}

element Stack::peek()
{
    if (isEmpty())
    {
```

```
        cout << "Magacinot e prazen i ne mozhe da se vadat elementi!" << endl;
        return{};
    }
    return array[top];
}

bool proverizagradi(string izraz, Stack &magacin)
{
    magacin.init();
    for (int i = 0; i < izraz.length(); i++) {
        if (izraz[i] == '{' || izraz[i] == '[' || izraz[i] == '(') {
            element e;
            e.info = izraz[i];
            magacin.push(e);
        }
        else if (izraz[i] == '}')
            if (magacin.peek().info == '{')
                magacin.pop();
            else return false;
        else if (izraz[i] == ']')
            if (magacin.peek().info == '[')
                magacin.pop();
            else
                return false;
        else if (izraz[i] == ')')
            if (magacin.peek().info == '(')
                magacin.pop();
            else
                return false;
    }
    if (magacin.isEmpty())
        return true;
    else
        return false;
}

int main()
{
    Stack magacin;
    string izraz;
    cout << "Vnesete aritmetichki izraz:" << endl;
    cin >> izraz;
    if (proverizagradi(izraz, magacin))
        cout << "Izrazot " << izraz << " ima balansirani zagradi";
    else
        cout << "Izrazot " << izraz << " nema balansirani zagradi";
}
```