

Рад са стеком (FIFO приступ)

Из задате улазне датотеке учитати податке у стек, при чему структура `pekara_st` садржи следећа поља:

- Назив пекаре (једна реч, до 16 карактера)
- Обим продаје пецива (природан цео број)
- Просечна цена пецива (позитиван реалан број)

Подразумевано, структура `pekara_st` садржи и поља за формирање стека.

На основу задате количине пецива које се направи на крају сваког дана `proizvodnja`, расподелити сва пецива пекарама на основу податка о обиму продаје пецива, тако да ако се појави вишак пецива, целокупан вишак расподелити пекари последњој на списку.

У случају да није произведено довољно пецива, приоритет да прве добију пецива имају пекаре које су се прве уписале на списак.

У задату излазну датотеку уписати податке у следећем формату:

- Назив пекаре (користити спецификатор `"%-10s"`)
- Добијена количина пецива (користити спецификатор `"%5d"`)
- Испуњеност обима продаје (једна реч до 8 карактера)

Испуњеност обима продаје представља разлику добијене количине пецива и обима продаје пецива, ако је она

- једнаки нули уписати „ОК”,
- ако је разлика негативна уписати „MANJAK”,
- ако је добијена позитивна разлика уписати „VIŠAK”

Пример позива програма:

```
./raspodela_peciva 21632 pekare.txt proizvodnja_i_prodaja.txt
```

Са податком `proizvodnja` = 21632 и задатим улазом у датотеци `pekare.txt`

Bondjorno	1853	59.62
Stara	925	42.85
Corsokak	8954	72.66
Kiflica	2152	22.63
Uzitak	3000	60.00
Bintel	4050	51.99
Mlin	698	18.56

И очекиваним излазом у датотеци `proizvodnja_i_prodaja.txt`:

Bondjorno	1853	OK
Stara	925	OK
Corsokak	8954	OK
Kiflica	2152	OK
Uzitak	3000	OK
Bintel	4050	OK
Mlin	698	OK

Пример позива програма:

```
./raspodela_peciva 2632 pekare.txt proizvodnja_i_prodaja.txt
```

Са податком **proizvodnja** = 2632 и задатим улазом у датотеци **pekare.txt**

Bondjorno	1853	59.62
Stara	925	42.85
Corsokak	8954	72.66
Kiflica	2152	22.63
Uzitak	3000	60.00
Bintel	4050	51.99
Mlin	698	18.56

И очекиваним излазом у датотеци **proizvodnja_i_prodaja.txt**:

Bondjorno	1853	OK
Stara	779	MANJAK
Corsokak	0	MANJAK
Kiflica	0	MANJAK
Uzitak	0	MANJAK
Bintel	0	MANJAK
Mlin	0	MANJAK

Пример позива програма:

```
./raspodela_peciva 21633 pekare.txt proizvodnja_i_prodaja.txt
```

И очекиваним излазом у датотеци **proizvodnja_i_prodaja.txt**:

Bondjorno	1853	OK
Stara	925	OK
Corsokak	8954	OK
Kiflica	2152	OK
Uzitak	3000	OK
Bintel	4050	OK
Mlin	699	VIŠAK

Пример решења:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_NAZIV 16 + 1

typedef struct pekara_st {
    char naziv[MAX_NAZIV];
    unsigned obimProdaje;
    double avgCena;
    struct pekara_st *sledeci;
} PEKARE;

void init_stack(PEKARE **);
PEKARE *create_item(const char *, const unsigned, const double);
unsigned push(PEKARE **, PEKARE *, FILE *, const unsigned);
PEKARE *pop(PEKARE **);
unsigned is_empty_stack(PEKARE *);
void empty_stack(PEKARE **);

void load_data(FILE *, FILE *, PEKARE **, const unsigned);
FILE *file_open(char *, char *);

unsigned raspodela_peciva(FILE *, PEKARE **, unsigned);

int main(int argn, char **args)
{
    if (argn != 4) {
        printf("USAGE: %s PROIZVODNJA IN_FILENAME OUT_FILENAME\n", args[0]);
        exit(11);
    }

    PEKARE *vrh;

    FILE *in = file_open(args[2], "r");
    FILE *out = file_open(args[3], "w");

    init_stack(&vrh);
    load_data(in, out, &vrh, atoi(args[1]));
    empty_stack(&vrh);

    fclose(in);
    fclose(out);

    return 0;
}

void init_stack(PEKARE **vrh) { *vrh = NULL; }

unsigned is_empty_stack(PEKARE *vrh)
{
    if(vrh == NULL)
        return 1;
    else
        return 0;
}

```

```
PEKARE *create_item(const char *naziv, const unsigned obimProdaje,
                    const double avgCena)
{
    PEKARE *new = (PEKARE *) malloc(sizeof(PEKARE));

    if(new == NULL) {
        printf("\nRAM allocation failed!\n\n");
        exit(42);
    }

    strcpy(new -> naziv, naziv);
    new -> obimProdaje = obimProdaje;
    new -> avgCena = avgCena;
    new -> sledeci = NULL;

    return new;
}

unsigned push(PEKARE **vrh, PEKARE *new, FILE *out, const unsigned proizvedeno)
{
    new -> sledeci = *vrh;
    *vrh = new;

    return raspodela_peciva(out, vrh, proizvedeno);
}

PEKARE *pop(PEKARE **vrh)
{
    if(is_empty_stack(*vrh))
        return NULL;

    PEKARE *tmp = *vrh;
    *vrh = (*vrh) -> sledeci;

    return tmp;
}

void empty_stack(PEKARE **vrh)
{
    while(*vrh != NULL) { pop(vrh); }
}

void load_data(FILE *in, FILE *out, PEKARE **vrh, const unsigned proizvedeno)
{
    PEKARE p;
    unsigned i = 0, j = 1;
    int ostatak = proizvedeno;

    while(fscanf(in, "%s %d %lf",
                p.naziv,
                &p.obimProdaje,
                &p.avgCena) != EOF) { i++; }

    fseek(in, 0, SEEK_SET);
    while(fscanf(in, "%s %d %lf",
                p.naziv,
                &p.obimProdaje,
                &p.avgCena) != EOF)
    {
```

```

        if(j != i)
            ostatak =
                push(vrh, create_item(p.naziv, p.obimProdaje, p.avgCena),
                    out, ostatak);
        else {
            if(ostatak > p.obimProdaje)
                fprintf(out, "%-10s %5d %-10s\n", p.naziv, ostatak, "VIŠAK");
            else
                ostatak =
                    push(vrh, create_item(p.naziv, p.obimProdaje, p.avgCena),
                        out, ostatak);
        }
        j++;
    }
}

FILE *file_open(char *name, char *mode)
{
    FILE *f = fopen(name, mode);

    if(f == NULL) {
        printf("\nFile doesn't exist!\n");
        exit(45);
    }
    return f;
}

unsigned raspodela_peciva(FILE *out, PEKARE **vrh, unsigned proizvedeno)
{
    if(is_empty_stack(*vrh))
        exit(EXIT_FAILURE);
    else {
        int ostatak, dobijeno;
        char status[10];
        proizvedeno = proizvedeno < 0 ? 0 : proizvedeno;

        if(proizvedeno >= (*vrh) -> obimProdaje) {
            ostatak = proizvedeno - ((*vrh) -> obimProdaje);
            dobijeno = (*vrh) -> obimProdaje;
            strcpy(status, "OK");
        }
        else if(proizvedeno == 0) {
            dobijeno = 0;
            strcpy(status, "MANJAK");
        }
        else {
            if(proizvedeno > 0) {
                dobijeno = ostatak;
                ostatak = 0;
            }
            strcpy(status, "MANJAK");
        }
        fprintf(out, "%-10s %5d %-10s\n", (*vrh) -> naziv, dobijeno, status);
        pop(vrh);

        return ostatak;
    }
}

```