

Vrátnik v kine dostal za úlohu zistiť, kto chodí do kina najčastejšie. Pomôžte mu a napíšte program, ktorý to spraví namiesto neho.

Na vstupe sú mená, každé tvorí jeden reťazec bez medzier. Na výstup napíšte meno, ktoré sa na vstupe vyskytovalo najčastejšie. Predpokladajte, že do kina chodí najviac 100 rôznych ľudí.

Ukážka vstupu:

Fero Tana Kubo
Maria Jozo Jozo
Kubo Marek Tana Hana
Kubo Tibor

Výstup pre ukážkový vstup:

Kubo

// uloha9-1.c -- Peter Plevko, 21.11.2019 13:10

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

struct Osoba
{
    char *meno;
    int pocet;
};

int main()
{
    struct Osoba o[100];
    int no = 0;
    char buf[100];
    int i, maxi = -1;
    while( scanf("%s",buf) > 0)
    {
        for (i = 0; i < no ; i++)
            if (!strcmp( buf, o[i].meno ))
            {
                o[ i ].pocet++;
                if ( o[maxi].pocet<o[i].pocet )
                    maxi = i;
                break;
            }
        if (i == no )
        {
            o[ i ].meno = strdup(buf);
            o[ i ].pocet = 1;
            no++;
            if (maxi < 0)
                maxi = 0;
        }
    }
    printf("%s\n", o[maxi].meno );

    return 0;
}
```

Kompilácia

1 Fero Tana Kubo Maria Jozo Kubo Marek Tana Hana Jozo

Štandardný výstup

1

Uvažujte implementáciu spájaného zoznamu využitím štruktúry POLOZKA, začiatok zoznamu je v premennej prvý.

Napište funkciu vlož(prvý,x,k), ktorá vloží do zoznamu položku obsahujúcu hodnotu x na k-te miesto v zozname, ktorého začiatok je v premennej prvý. Pozíciu k počítame od 0, kedy vložíme na začiatok zoznamu. Ak k je väčšie ale rovné ako aktuálny počet prvkov v zozname, funkcia vloží položku na koniec zoznamu. Ak premenná prvý je NULL, tak zoznam je prázdny.

Riešenie použite v programe, ktorý načíta zo vstupu hodnoty cislo, k a spájaný zoznam funkciou nacitaj(), vloží hodnotu cislo na k-te miesto, a vypíše výsledný zoznam funkciou vypis(). Funkcie nacitaj() a vypis() nemusíte implementovať.

Ukážka zoznamu:

6 -> 4 -> 2 -> 1

Ukážka zoznamu po volaní vlož(prvý,9,2):

6 -> 4 -> 9 -> 2 -> 1

```
1 // uloha9-2.c -- Peter Plevko, 19.11.2019 20:55
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct Polozka
7 {
8     int cislo;
9     struct Polozka *dalsi;
10 } POLOZKA;
11
12 POLOZKA *nacitaj();
13
14 void vypis(POLOZKA *prvy);
15
16 void vlož(POLOZKA **prvy, int cislo, int k)
17 {
18     POLOZKA *paPocetprvkovvzozname = NULL;
19     int pocetprvkov=0;
20     paPocetprvkovvzozname=*prvy;
21     while (paPocetprvkovvzozname != NULL)
22     {
23         pocetprvkov++;
24         paPocetprvkovvzozname = paPocetprvkovvzozname->dalsi;
25     }
26
27     POLOZKA *ptr = (POLOZKA*)malloc(sizeof(POLOZKA));
28
29     ptr->cislo=cislo;
30
31     int i;
32     POLOZKA *temp=*prvy,*novyhead=NULL;
33     novyhead=*prvy;
34
35     if (*prvy == NULL)
36     {
37         ptr->dalsi=NULL;
38         *prvy=ptr;
39     }
40     else
41     {
42         if (k==0)
43         {
44             ptr->dalsi=*prvy;
45             *prvy=ptr;
46         }
47         else
48         {
49             while (i<k-1)
50             {
51                 temp=temp->dalsi;
52                 i++;
53             }
54             ptr->dalsi=temp->dalsi;
55             temp->dalsi=ptr;
56         }
57     }
58 }
```

```
1 // uloha9-2.c -- Peter Plevko, 19.11.2019 20:55
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct Polozka
7 {
8     int cislo;
9     struct Polozka *dalsi;
10 } POLOZKA;
11
12 POLOZKA *nacitaj();
13
14 void vypis(POLOZKA *prvy);
15
16 void vlož(POLOZKA **prvy, int cislo, int k)
17 {
18     POLOZKA *paPocetprvkovvzozname = NULL;
19     int pocetprvkov=0;
20     paPocetprvkovvzozname=*prvy;
21     while (paPocetprvkovvzozname != NULL)
22     {
23         pocetprvkov++;
24         paPocetprvkovvzozname = paPocetprvkovvzozname->dalsi;
25     }
26
27     POLOZKA *ptr = (POLOZKA*)malloc(sizeof(POLOZKA));
28
29     ptr->cislo=cislo;
30
31     int i;
32     POLOZKA *temp=*prvy,*novyhead=NULL;
33     novyhead=*prvy;
34
35     if (*prvy == NULL)
36     {
37         ptr->dalsi=NULL;
38         *prvy=ptr;
39     }
40     else
41     {
42         if (k==0)
43         {
44             ptr->dalsi=*prvy;
45             *prvy=ptr;
46         }
47         else
48         {
49             while (i<k-1)
50             {
51                 temp=temp->dalsi;
52                 i++;
53             }
54             ptr->dalsi=temp->dalsi;
55             temp->dalsi=ptr;
56         }
57     }
58 }
```

```
1 5 4
2 3 9 6
```

Uvažujte implementáciu spájaného zoznamu využitím štruktúry POLOZKA, začiatok zoznamu je v premennej prvý.

Napište funkciu `zmaz(prvy,k)`, ktorá odstráni k-tu položku zo zoznamu, ktorého začiatok je v premennej `prvy`. Pozíciu k počítame od 0, kedy odstránime prvú položku. Ak k je väčšie ale rovné ako aktuálny počet prvkov v zozname, funkcia zoznam nebude upravovať. Prázdny zoznam je reprezentovaný hodnotou NULL.

Riešenie použite v programe, ktorý načíta zo vstupu hodnotu k a spájaný zoznam funkciou `nacitaj()`, odstráni k-tu položku, a vypíše výsledný zoznam funkciou `vypis()`. Funkcie `nacitaj()` a `vypis()` nemusíte implementovať.

Ukážka zoznamu:

6 -> 4 -> 2 -> 1

Ukážka zoznamu po volaní `zmaz(prvy,1)`:

6 -> 2 -> 1

```
1 // uloha9-3.c -- Peter Plevko, 22.11.2019 15:35
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct Polozka
7 {
8     int cislo;
9     struct Polozka *dalsi;
10 } POLOZKA;
11
12 POLOZKA *nacitaj();
13 void vypis(POLOZKA *prvy);
14
15 void zmaz(POLOZKA **prvy, int k)
16 {
17
18     if (*prvy == NULL)
19         return;
20
21     POLOZKA* temp = *prvy;
22
23
24
25     if (k == 0)
26     {
27         *prvy = temp->dalsi;
28         free(temp);
29         return;
30     }
31
32
33     for (int i=0; temp!=NULL && i<k-1; i++)
34         temp = temp->dalsi;
35
36     if (temp == NULL || temp->dalsi == NULL)
37         return;
```

1 0
2 5

1

Uvažujte implementáciu spájaného zoznamu využitím štruktúry POLOZKA, začiatok zoznamu je v premennej prvy.

Napište funkciu `zmaz_delitelne(prvy,k)`, ktorá odstráni všetky položky deliteľné `k` zo zoznamu, ktorého začiatok je v premennej `prvy`. Prázdny zoznam je reprezentovaný hodnotou `NULL`.

Riešenie použite v programe, ktorý načíta zo vstupu hodnotu `k` a spájaný zoznam funkciou `nacitaj()`, odstráni položky deliteľné `k`, a vypíše výsledný zoznam funkciou `vypis()`. Funkcie `nacitaj()` a `vypis()` nemusíte implementovať.

Ukážka zoznamu:

6 -> 4 -> 3 -> 2 -> 1

Ukážka zoznamu po volaní `zmaz_delitelne(prvy,2)`:

3 -> 1

```
1 // uloha9-4.c -- Peter Plevko, 22.11.2019 15:36
```

```
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct Polozka
7 {
8     int cislo;
9     struct Polozka *dalsi;
10 } POLOZKA;
11
12 POLOZKA *nacitaj();
13 void vypis(POLOZKA *prvy);
14
15 void zmaz_delitelne(POLOZKA **prvy, int k)
16 {
17
18     POLOZKA *predosly, *aktualny;
19
20     while (*prvy != NULL && ((*prvy)->cislo) % k == 0)
21     {
22         predosly = *prvy;
23
24         *prvy = (*prvy)->dalsi;
25
26         free(predosly);
27
28     }
29
30     predosly = NULL;
31     aktualny = *prvy;
32
33     while (aktualny != NULL)
34     {
```

```
1 // uloha9-4.c -- Peter Plevko, 22.11.2019 15:36
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct Polozka
7 {
8     int cislo;
9     struct Polozka *dalsi;
10 } POLOZKA;
11
12 POLOZKA *nacitaj();
13 void vypis(POLOZKA *prvy);
14
15 void zmaz_delitelne(POLOZKA **prvy, int k)
16 {
17
18     POLOZKA *predosly, *aktualny;
19
20     while (*prvy != NULL && ((*prvy)->cislo) % k == 0)
21     {
22         predosly = *prvy;
23
24         *prvy = (*prvy)->dalsi;
25
26         free(predosly);
27
28     }
29
30     predosly = NULL;
31     aktualny = *prvy;
32
33     while (aktualny != NULL)
34     {
```

```
1 2
2 6 4 3 2 1
```

```
1
```

Uvažujte implementáciu spájaného zoznamu využitím štruktúry POLOZKA, začiatok zoznamu je v premennej prvý.

Napište funkciu `otoc(prvy)`, ktorá otočí poradie položiek v zozname, ktorého začiatok je v premennej `prvy`. Prázdny zoznam je reprezentovaný hodnotou `NULL`. Funkcia neupravuje hodnoty položiek ale mení prepojenie položiek.

Riešenie použite v programe, ktorý načíta zo vstupu spájaný zoznam funkciou `nacitaj()`, otočí položky zoznamu, a vypíše výsledný zoznam funkciou `vypis()`. Funkcie `nacitaj()` a `vypis()` nemusíte implementovať.

Ukážka zoznamu:

6 -> 4 -> 3 -> 2 -> 1

Ukážka zoznamu po volaní `otoc(prvy)`:

1 -> 2 -> 3 -> 4 -> 6

```
1 // uloha9-5.c -- Peter Plevko, 22.11.2019 15:36
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 typedef struct Polozka
7 {
8     int cislo;
9     struct Polozka *dalsi;
10 } POLOZKA;
11
12 POLOZKA *nacitaj();
13 void vypis(POLOZKA *prvy);
14
15 void otoc(POLOZKA **prvy)
16 {
17     POLOZKA* prev = NULL;
18     POLOZKA * current = *prvy;
19     POLOZKA * next = NULL;
20     while (current != NULL) {
21         next = current->dalsi;
22
23         current->dalsi = prev;
24         prev = current;
25         current = next;
26     }
27     *prvy = prev;
28 }
29
30
31
32
33
34 int main()
35 {
36     POLOZKA *prvy;
37     prvý = nacitaj(); // načítaj zoznam
```

Uvažujte implementáciu spájaného zoznamu využitím štruktúry POLOZKA, začiatok zoznamu je v premennej prvy.

Napište funkciu vypis(prvy), ktorá vypíše položky zoznamu, ktorého začiatok je v premennej prvy. Prázdny zoznam je reprezentovaný hodnotou NULL. Pre prázdny zoznam funkcia vypíše správu Prázdny zoznam

Riešenie použite v programe, ktorý načíta zo vstupu spájaný zoznam funkciou nacitaj() a vypíše položky zoznamu funkciou vypis(prvy). Funkciu nacitaj() nemusíte implementovať.

Ukážka zoznamu:

6 -> 4 -> 3 -> 2 -> 1

Výpis zoznamu volaním vypis(prvy):

6 -> 4 -> 3 -> 2 -> 1

```
// uloha9-6.c -- Peter Plevko, 29.11.2019 23:28
```

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

```
typedef struct Polozka
{
    int cislo;
    struct Polozka *dalsi;
} POLOZKA;
```

```
POLOZKA *nacitaj();
```

```
void vypis(POLOZKA *prvy)
{
    if ( prvy==NULL )
        printf("Prázdny zoznam\n");
    else
    {
        printf("%d", prvy->cislo );
        if ( prvy->dalsi != NULL )
        {
            printf(" -> ");
            vypis( prvy->dalsi );
        }
    }
}
```

```
int main()
{
    POLOZKA *prvy;
    prvy = nacitaj(); // nacitaj zoznam
    vypis(prvy);
    return 0;
}
```

Zadanie	Testy	uloha9-7.c	Štandardný vstup
<p>Uvažujte implementáciu spájaného zoznamu využitím štruktúry POLOZKA, začiatok zoznamu je v premennej prvy.</p>		<pre>// uloha9-7.c -- Peter Plevko, 29.11.2019 23:34 #include <stdio.h> #include <stdlib.h> typedef struct Polozka { int cislo; struct Polozka *dalsi; } POLOZKA; POLOZKA *nacitaj(); void vypis_opacne(POLOZKA *prvy) { if (prvy==NULL) printf("Prazdny zoznam\n"); else { if (prvy->dalsi != NULL) { vypis_opacne(prvy->dalsi); printf(" -> "); } printf("%d", prvy->cislo); } } int main() { POLOZKA *prvy; prvy = nacitaj(); // nacitaj zoznam vypis_opacne(prvy); return 0; }</pre>	<div>1</div>
<p>Napište funkciu vypis_opacne(prvy), ktorá od konca vypíše položky zoznamu, ktorého začiatok je v premennej prvy. Prázdny zoznam je reprezentovaný hodnotou NULL. Pre prázdny zoznam funkcia vypíše správu Prazdny zoznam</p>		<div>X</div>	
<p>Riešenie použite v programe, ktorý načíta zo vstupu spájaný zoznam funkciou nacitaj() a vypíše položky zoznamu funkciou vypis_opacne(prvy). Funkciu nacitaj() nemusíte implementovať.</p>		<div>X</div>	
<p>Ukážka zoznamu: 6 -> 4 -> 3 -> 2 -> 1 Výpis zoznamu volaním vypis_opacne(prvy): 1 -> 2 -> 3 -> 4 -> 6</p>		<div>X</div>	
		<div>X</div>	
		<div>X</div>	
		<div>X</div>	<div>Štandardný výstup</div>
		<div>X</div>	<div>1</div>
		<div>X</div>	
		<div>X</div>	
		Kompilácia	