



Задачи

Изготвил:
гл.ас. д-р Нора Ангелова

Дадени са естествени числа n и m .

Предполага се, че m човека са наредени в кръг и всеки от тях е получил пореден номер от 1 до m .

Отброява се n -тият човек и се отстранява от кръга. Започвайки от следващия човек ($n+1$) се отброява n -тият човек и се отстранява. Този процес продължава до отстраняване на всички от кръга.

Да се напише програма, която извежда номерата на отстранените в реда на отстраняването им.

```
typedef  CirList<int>  IntCir;
```

```
void create(int m, IntCir &L);
```

```
void deleteN(int n, IntCir L);
```

```
int main()
```

```
{
```

```
    IntCir L;
```

```
    int m, n;
```

```
    cout << "m: "; cin >> m;
```

```
    create(m, L);
```

```
    L.print();
```

```
    cout << "n: "; cin >> n;
```

```
    deleteN(n, L);
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

```
void create(int m, IntCir &L)
{
    for (int i = 1; i <= m; i++)
        L.ToEnd(i);
}

void deleteN(int n, IntCir L)
{
    L.IterStart();
    elem_cir<int> *p = L.Iter(), *q;

    while(p != p->link)
    {
        q = p;
        for(int i=1; i <= n-1; i++)
            q = q->link;


        p = q->link;
        int x;
        L.DeleteElem(q, x);
        cout << x << " ";
    }
    cout << p->inf << endl;
}
```



Да се напише шаблон на функцията от по-висок ред `accumulate` за циклични свързани списъци.

```
template <typename T>
T accumulate(T (*op) (T, T), T null_val,
CirList<T>& L)
{
    T s = null_val;
    L.IterStart();
    elem_cir<T> *p = L.Iter();
    while(p)
    {
        s = op(s, p->inf);
        p = L.Iter(); // p=p>link зацикляне!
    }


    return s;
}
```



Да се напише шаблон на функция от по-висок
ред `filter` за циклични свързани списъци

```
template <typename T>
CirList<T> filter(bool (* pred) (T), CirList<T> &L)
{
    L.IterStart();
    elem_cir<T> *p = L.Iter();
    CirList<T> filteredList;
    while (p)
    {
        if (pred(p->inf))
            filteredList.ToEnd(p->inf);
        p = L.Iter();
    }

    return filteredList;
}
```

Да се напише програма, която създава цикличен свързан списък, изграден от циклични свързани списъци от числа, след което конструира цикличен свързан списък, резултат от конкатенацията на цикличните свързани списъци, съставлящи дадения.

```
typedef CirList<int> IntList;
typedef CirList<IntList> IntListList;

void create(int n, IntListList & resultList)
{
    for(int i=0; i<n; i++)
    {
        IntList list;
        int length;

        cout << "Length of the list: ";
        cin >> length;

        for(int j=0; j<length; j++)
        {
            int x;
            cout << "elem: ";
            cin >> x;
            list.ToEnd(x);
        }

        resultList.ToEnd(list);
    }
}
```

```
IntList concat(IntList list1, IntList list2)
```

```
{  
    IntList list;  
    list1.IterStart();  
    list2.IterStart();  
    elem_cir<int> *p = list1.Iter();  
    elem_cir<int> *q = list2.Iter();
```

```
    while(p)  
    {  
        list.ToEnd(p->inf);  
        p = list1.Iter();  
    }
```

```
    while(q)  
    {  
        list.ToEnd(q->inf);  
        q = list2.Iter();  
    }
```

```
    return list;  
}
```

Вариант 2

```
IntList list = list1;  
list2.IterStart();  
elem_cir<int> *q = list2.Iter();
```

```
while(q)  
{  
    list.ToEnd(q->inf);  
    q = list2.Iter();  
}
```

```
return list;  
}
```

```
int main()
{
    IntListList list;
    create(3, list);
    IntList emptyList;
    IntList concatList = accumulate(concat, emptyList, list);
    concatList.print();

    system("pause");
    return 0;
}
```

Свързан списък, съдържащ $2n$ цели числа $a_1, a_2, \dots, a_{2n-1}, a_{2n}$,

е представен чрез две връзки.

Да се напише функция, която намира:

- $S = a_1 * a_{2n} + a_2 * a_{2n-1} + a_n * a_{n+1}$
- $M = \max(\min(a_1, a_{2n}), \min(a_2, a_{2n-1}), \dots, \min(a_n, a_{n+1}))$

Свързан списък, съдържащ $2n$ цели числа
 $a_1, a_2, \dots, a_{2n-1}, a_{2n}$,

е представен чрез две връзки.

Да се напише функция, която намира:

- $S = a_1 * a_{2n} + a_2 * a_{2n-1} + \dots + a_n * a_{n+1}$

```
typedef DList<int> IntDList;

int sum(IntDList & list)
{
    list.IterStart();
    elem_link2<int> *p = list.IterNext();
    list.IterEnd();
    elem_link2<int> *q = list.IterPrev();

    int n = list.length()/2;
    int s = 0;
    for(int i=1; i<=n; i++)
    {
        s = s + p->inf * q->inf;
        p = list.IterNext();
        q = list.IterPrev();
    }

    return s;
}
```

Свързан списък, съдържащ $2n$ цели числа
 $a_1, a_2, \dots, a_{2n-1}, a_{2n}$,

е представен чрез две връзки.

Да се напише функция, която намира:

- $M = \max(\min(a_1, a_{2n}), \min(a_2, a_{2n-1}), \dots, \min(a_n, a_{n+1}))$


```
int maximin(IntDList & list)
{
    list.IterStart();
    elem_link2<int> *p = list.IterNext();
    list.IterEnd();
    elem_link2<int> *q = list.IterPrev();

    int n = list.length()/2;
    int max = p->inf <= q->inf ? p->inf : q->inf;

    for(int i=1; i<=n-1; i++)
    {
        p = list.IterNext();
        q = list.IterPrev();
        int min = p->inf <= q->inf ? p->inf : q->inf;
        if (min > max) max = min;
    }

    return max;
}
```

Полиномът

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

е представен чрез свързан списък.

Ако a_n е 0, елементът не се включва в списъка.

- Въвеждане
- Извеждане
- Стойност
- Производна
- Сума на полиноми
- Произведение на полином и едночлен
- Произведение на два полинома