

ПРОГРАМИРАНЕ НА ЧИСЛОВИ РЕДИЦИ

Ивайло Старибратов, Цветана Димитрова
Образцова математическа гимназия „Акад. Кирил Попов”

„Колкото човек е по-близо, толкова по-малко вижда“
Зрителна измама, филм на Луи Летерие

Резюме. Разработена е система от опорни задачи, свързана с приложението на числови редици от математиката в информатиката. За придобиване на знания и умения у учениците за циклични алгоритми се използват числови редици. Темата е насочена към състезателна група Е по информатика (10 – 11-годишни ученици).

Keywords: informatics for competitions, methodology, programing for fourth graders

През последните пет години интересът към програмирането все повече се засилва от страна на началното училище. Учениците започват да се занимават с информатика още от трети клас в извънкласна форма на обучение. Примерната учебна програма за обучение по програмиране в ИКРИ е направена за V клас и по нея се обучават всички начинаещи (Infos, 2014). Периодът на обучение става по-дълъг и това води до увеличаване на броя часове и добавяне на нови теми. Подготовката на учениците изисква много добре разработена методика. Тази методика трябва да стимулира алгоритмичното мислене и да застъпва игровия елемент.

Разработената тема е подходяща за ученици от IV и V клас в извънкласна форма на обучение по програмиране в частта „Циклични алгоритми“.

Числовите редици не се изучават в явен вид на тази възраст, но учениците имат понятие за тях. По учебен план темата се изучава в XI клас по предмета математика. Това не прави темата по-трудна. Обикновено темите, заложиени в предмет, който се изучава в извънкласна форма, изпреварват темите, заложиени в редовна форма на обучение.

Състояние на проблема

Решаването на задачи по информатика способства за развитие на логическото мислене, формиране на интелектуални умения за работа с разнородна по характер информация.

Задачите имат 5 основни функции

– методическа;

– дидактическа;

– организираща и управляваща;

– развиваща – съдейства за умственото развитие на учениците, за формиране на определени качества на мисленето: гъвкавост, логичност, системност, обобщеност, самостоятелност, умения за взимане на решения и др.;

– възпитателна – задачите в обучението по информатика съдействат за формиране на определени нравствени качества, възгледи, естетически усещания, дисциплинираност, висока организаторска и информационна култура, деловитост, умения за планиране, прецизност при работа с компютърна техника, умения за прилагане на знанията в практиката.

Тя произтича от основните цели и задачи в обучението по информатика, които служат за подготовка, въвеждане и затвърждаване на знанията за алгоритмите и изучаваните системи от приложното програмно осигуряване, предвидени за усвояване и формиране на умения и навици за работа (Дурева, 2003).

След изучаване на същността на разклонените алгоритми, както и операторите, които ги осъществяват, се преминава към цикличните алгоритми. Разработените учебни помагала до този момент по тази тема помагат на учениците да придобият знания и умения, свързани с откриването на повтарящи се действия, да променят и съставят такива, да определят условия за повтарящи се действия, да разбират необходимостта от използване на този вид конструкции.

Разглеждат се следните задачи: Да се отпечата даден знак 100 пъти, да се отпечата целите числа от 1 до 100, да се изведе сума на числа, въведени от клавиатурата, да се изведе най-голямото число от въведените от клавиатурата, пресмятане на факториел. Задачи за упражнение: Да се изведат всички цели числа от n до 1, да се изведат всички цели числа от 5 до 100, от 20 до 35, да се изведат квадратите на всички цели положителни цели числа от 10 до b ($b \leq 10$), да се изведат третите степени на всички цели положителни числа от a до 5 ($a \leq 5$), всички цели положителни числа от a до b ($a \leq b$), да се отпечата в стълб таблицата за умножение със 7, да се напише програма, с която се пресмята сумата $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{10}$. Запознаване с редицата на Фибоначи. Дава се решение на задачата да се отпечата последователно числата от редицата, като се започне от третото, с помощта на оператора `while`. Отпечатване на сумата на последователно въведени числа, като е зададено условие за край на въвеждането (Келеведжиев & Дженкова, 2004).

Като опорни задачи за подготовка на изявени ученици за участие в олимпиади по информатика се явяват такива, в които се изисква да се напише програма, коя-

то пресмята сумите $1 + \frac{1}{2} + \dots + \frac{1}{100}$, както и $1 + 2.3 + 3.4.5 + 4.5.6.7 + \dots$ (Гъров, 2008).

В литературата се разглеждат някои задачи като: последователно въведени числа от клавиатурата, определя се знак за край на въвеждането; програма, която трябва да изведе сумата от въведените числа, както и задача, която изисква програмата да изведе броя на въведените числа, произведението на въведените числа, броя на въведените четни положителни и максималното такова; задача, в която се разглеждат числа в интервал и задача `squares.cpp`, в която се въвежда цяло число n и програмата трябва да изведе квадратите на целите числа в интервала от 1 до n (Йовчева & Иванова, 2007).

В ръководство за упражнения по програмиране, разглеждайки операторите, осъществяващи циклични алгоритми, се използват задачи, в които се изисква последователно въвеждане на числа. Програмата трябва да извежда сума, произведение и средноаритметично на въведените числа (Христова & Григорова, 2011).

Приведените факти показват, че при запознаване на учениците с цикличните алгоритми и оператори те неявно се запознават с редицата на естествените числа, обработват както до n -тото число на редицата, така и в интервал, познават редицата на Фибоначи, намират сума на редица от дробни числа.

Представена е система от типови задачи, в които се изисква образуване на числови редици по определено правило, както и пресмятане на сума и произведение на членовете на така образуваните редици. Освен че учениците ще придобият знания за образуване на различни редици от числа, те ще придобият и техника при решаване на този тип задачи от циклични алгоритми. Представено е едно по-различно решение за съставяне на редицата на Фибоначи, както и една съществителна авторска задача за самостоятелна работа.

Системата от задачи следва модела на асоциативно учене, който е: подготовка за усвояване на нов материал; съобщаване на нови факти и сведения; сравняване и абстрахиране на свойства и отношения; обобщаване; прилагане на усвоени знания (Андреев, 2003).

Педагогически подход

Трябва да се отбележи, че понятията „числова редица“ от математиката и „редица от числа“ в информатиката (едномерен масив) са две отделни понятия. Едно неформално определение за числова редица в математиката има следния вид: множество от числа, всяко от които има свой номер n , където $n = 1, 2, 3, \dots$, се нарича числова редица. Казано с други думи, числовата редица е поредица от числа, която следва някаква логика. Например редицата 1, 3, 5, 7, 9, и т.н. В нея логиката е, че всяко следващо число е с 2 по-голямо от предишното. Тези числа

от числовата редица се наричат членове. За разглежданите редици от числа в типовете задачи може да се определи общ член. Нека a_1 – е първият, а a_2 – вторият член на редицата. Тогава, ако с a_n се отбележи общият член на редицата, то ще е в сила равенството $a_n = 2 * n - 1$. Ако $n = 4$, за a_4 се получава следното равенство: $a_4 = 2 * 4 - 1 = 7$.

Разяснява се понятието „аритметична прогресия“ и формулите, по които може да се пресметнат общият член и сумата от членовете на такава прогресия, както и още няколко формули, които ще направят алгоритмите по-бързи и ще помогнат за решаването на задачи, в които се изисква да се направи „сух пробег“:

Формула	Предназначение
$\frac{n * (n + 1)}{2}$	Намира сумата на първите n последователни цели числа: 1, 2, 3, ...
$\frac{(a + b) * (b - a + 1)}{2}$	Намира сумата на n последователни цели числа в интервал $[a, b]$.
$1 + 3 + 5 + + (2 * k - 1) = k^2$, при $k = 1, 2, 3,$	Намира сумата на първите n нечетни цели числа: 1, 3, 5, 7, 9,

Таблица 1

Дидактическа система от задачи

Първи тип задачи (1ТЗ)

Извеждане на зададени с множество числови редици. Намиране на сума и произведение до n -тия член на редицата.

1ТЗ.1 Напишете програма, която извежда следните редици от цели числа на един ред през един интервал (до 10-я член на редицата):

А) $P = \{1, 2, 3, 4, 5, \dots\}$; Б) $P = \{1, 3, 5, 7, 9, \dots\}$; В) $P = \{2, 4, 6, 8, 10, \dots\}$;

Г) $P = \{1, 4, 7, 10, 13, \dots\}$; Д) $P = \left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots\right\}$.

1ТЗ.2 Напишете програма, която извежда стойността на n -тия член на редиците от 1ТЗ.1. Числото n се въвежда от клавиатурата.

А) при $n = 8$; Б) при $n = 9$; В) при $n = 10$.

1ТЗ.3 Напишете програма, която извежда сумата до n -тия член на редиците от 1ТЗ.1. Числото n се въвежда от клавиатурата.

А) при $n = 5$; Б) при $n = 6$; В) при $n = 7$.

1ТЗ.4 Напишете програма, която извежда произведението до n -тия член на редиците от 1ТЗ.1. Числото n се въвежда от клавиатурата.

А) при $n = 3$; Б) при $n = 4$; В) при $n = 5$.

В 1ТЗ.1 числовите редици са дадени. За решението на задачата трябва да се открие зависимостта между членовете на редицата. Формулата, в която участва индексната променлива, управляваща цикъла, ще помогне за извеждане на съответната редица. В 1ТЗ.2, използвайки формулите от ТЗ1.1, се извеждат стойностите при съответните примерни входове. В 1ТЗ.3 и 1ТЗ.4 се пресмята сумата и произведението до n -тия член на редиците от 1ТЗ.1. В 1ТЗ.3 се използва функцията `int NOD (int a, int b)`, за да се пресметне сумата на членовете в редицата, записана в 1ТЗ.1Д. Сумата и произведението от членовете на редицата 1ТЗ.1Д се извеждат като обикновена дроб с числител и знаменател (например $\frac{1}{5040}$), а не дробно число с десетична запетая.

Втори тип задачи (2ТЗ)

Извеждане на зададени с множество числови редици в определен интервал. Намиране на сума и произведение на членовете в зададения интервал.

2ТЗ.1 Напишете програма, която извежда следните редици от цели числа на един ред през един интервал:

- А) $P = \{13, 14, 15, 16, 17, 18, 19\}$; Б) $P = \{3, 5, 7, 9, 11, 13, 15\}$;
В) $P = \{6, 8, 10, 12, 14, 16, 18\}$; Г) $P = \{30, 2, 20, 15, 10, 5, 1\}$;
Д) $P = \{100, 90, 80, 70, 60, 50, 40\}$.

2ТЗ.2 Напишете програма, която извежда сумата от членовете на редиците от 2ТЗ.1, като интервалът $[a, b]$, където $a \leq b$, се задава от клавиатурата. С a и b са означени номерата на членовете на редицата. Например $[3, 7]$ означава от третия до седмия член на редицата 2ТЗ.1А. Всяка подточка съответства на редицата, записана със същата подточка от 2ТЗ.1.

- А) $[3, 7]$; Б) $[4, 7]$; В) $[1, 6]$; Г) $[1, 4]$; Д) $[3, 6]$.

2ТЗ.3 Напишете програма, която извежда произведението от членовете на редиците от 2ТЗ.1 като интервалът $[a, b]$, където $a \leq b$, се задава от клавиатурата. Всяка подточка съответства на редицата, записана със същата подточка от 2ТЗ.1.

- А) $[3, 7]$; Б) $[4, 7]$; В) $[1, 6]$; Г) $[1, 4]$; Д) $[3, 6]$.

В 2ТЗ.1 числовите редици са зададени в интервал. Първите три подточки са частен случай на първите три от 1ТЗ.1, а подусловия Г) и Д) се образуват по нови

правила, като редиците са в низходящ ред. 2ТЗ.2 и 2ТЗ.3 са аналогични. За решението им се използват правилата за образуване на редици от 2ТЗ.1. Образуват се сумата и произведението в интервала с помощта на оператора if.

Трети тип задачи (3ТЗ)

Извеждане на числови редици по определено правило. Намиране на сума и произведение на членовете на редицата.

3ТЗ.1 Напишете програма, която извежда първите пет члена на редица от числа с общ член, равен на $(n \geq 1)$:

А) $2 * n - 1$; Б) $\frac{2 * n}{n + 1}$; В) $(-1)^n * \frac{n}{n + 1}$; Г) $n!$.

3ТЗ.2 Напишете програма, която извежда първите пет члена на редица от числа с общ член, равен на $(n \geq 1)$:

А) $5 * (-1)^n$; Б) $\begin{cases} 0, & \text{при } n - \text{нечетно} \\ \frac{2}{n}, & \text{при } n - \text{четно} \end{cases}$; В) $(-1)^{n+1} * 2^n$.

Този тип задачи изисква по зададена формула за общия член на редицата да се изведат първите няколко члена на редицата. Редиците в 3ТЗ.2 са от вида $2, -2, 2, -2, \dots$ или този $0, 1, 0, \frac{1}{2}, \dots$.

Четвърти тип задачи (4ТЗ)

„Сух пробег“ на фрагмент от програма.

4ТЗ.1 Определете какво ще бъде отпечатано в резултат от изпълнението на фрагмента:

```
int sum, i;  
i=1;  
sum=0;  
while (i<6)  
{  
    sum=sum+i;  
    i=i+1;  
}  
cout<<sum<<endl;
```

4ТЗ.2 Определете какво ще бъде отпечатано в резултат от изпълнението на фрагмента:

```
int a=3, b=6, i, sum=0;
for (i=a; i<=b; i++)
    sum = sum+i;
cout<<sum<<endl;
```

4ТЗ.3 Определете какво ще бъде отпечатано в резултат от изпълнението на фрагмента:

```
int a[100], i, n= 5, sum = 0;
a[0]=1;
a[1]=3;
for(i=2; i<=n; i++)
{
    a[i]=a[i-1]+2;
    cout<<a[i]<<" ";
}
cout<<endl;
for (i=0; i<=n; i++)
{
    sum = sum + a[i];
}
cout<<sum<<endl;
```

В този тип задачи се прави „сух пробег“ и се изисква съобразителност от страна на решаващия. Използването на формулите от математиката ще ускорят времето за решаването им.

Следващата задача представя едно по-различно решение за съставяне на редицата на Фибоначи.

Задача: Напишете програма, която извежда редицата на Фибоначи. Правило за образуване на редицата: първите два елемента на редицата са 1 и 1, а всеки следващ се получава като сума от предходните два. На стандартния вход се въвежда едно число n ($n < 25$) – до кой елемент да се отпечата редицата. На стандартния изход – редицата от числа, разделени с един интервал.

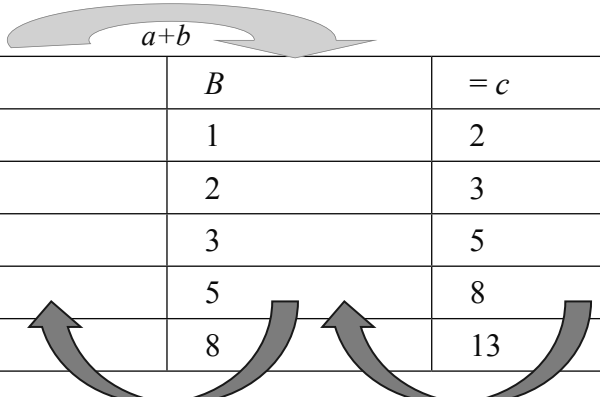
Анализ на задачата:

Първият и вторият елемент имат стойност 1. Третият е равен на първия плюс втория, т.е. $1 + 1 = 2$. Четвъртият елемент е равен на сбора от втория и третия, т.е. $1 + 2 = 3$.

Петият е равен на сбора от третия и четвъртия, т.е. $2 + 3 = 5$, и т.н.

Редицата е: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

Попълва се таблица със стойностите на променливите:



A	B	$= c$
1	1	2
1	2	3
2	3	5
3	5	8
5	8	13

Таблица 2.

Получени изрази от таблица 2:

$a = 1, b = 1;$ //първите два члена на редицата

$c = a + b;$ //в c ще натрупваме сумата, като в a ще помним последната стойност на b , а в c – последната стойност на b , т.е. $a = b; b = c;$

В масив се съхраняват текущите стойности на променливата c .

Решение на езика C++, среда Code::Blocks 10.05

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    long a=1, b=1, i, c, n, masiv[100];
```

```
    cin>>n;
```

```
    for (i=2; i<n; i++)
```

```
    {
```

```
        c = a + b;
```

```
        a = b;
```

```
        b = c;
```

```
        masiv[i] = c; //запазване на текущите стойности на  $c$  в масив
```

```
    }
```

```
    masiv[0]=1;
```

```
    masiv[1]=1;
```

```
    for (i=0; i < n; i++)
```

```
cout<<masiv[i]<<" ";  
cout<<endl;  
return 0;  
}
```

Представената по-долу задача е състезателна и е предназначена за самостоятелна работа на учениците.

АПТЕКАРЯТ С ВЪЛШЕБНИТЕ КОЛБИ

Имало едно време един аптекар. Той продавал всякакви лекарства и отвари, които приготвял сам. Освен че бил много вещ в тайните на лековете, той бил и много прецизен в съотношенията, които използвал. Един ден на местния панаир си купил една огромна кутия с N колби, които били еднакви по форма и цвят, но всяка следваща била по-голяма от предишните. На дъното на всяка от тях били изписани последователно числата от 1 до N . Аптекарят побързал да попита странния продавач как да разбере колко литра се събират във всяко от бурканчетата. Продавачът му казал: „Тези бурканчета са вълшебни. В първата колба никога не поставяй нищо, тъй като колкото и да наливаш в нея, тя ще е винаги празна. Всяка следваща колба събира толкова литри, колкото, ако към номера на предишната колба прибавиш литрите, които събира предходната колба. И помни – счупиш ли колба с нечетен номер, няма да се случи нищо, но счупиш ли колба с четен номер, всички колби с четни номера ще изчезнат“.



Не щеш ли, докато мъкнел огромната кутия към къщи, аптекарят се спънал, кутията се залюляла и една от колбите се изтъркулила и се счупила. В миг всички колби заедно с местата си с четни номера изчезнали. Аптекарят започнал да се вайка: „Защо трябваше да падне точно колба с четен номер...“. Но накрая се прибрал и си казал „Е, не е чак толкова лошо. Все пак ми останаха всички нечетни колби“.

Години наред, пък и до ден-дневен наливал приготвената отвара в колба, която му се виждала приблизително равна на количеството, което е приготвил. За нещастие отварите му взаимодействат с остатъчния въздух в колбите и се развалят. За да не се случва това, аптекарят трябва да намери най-малката колба, която събира количеството приготвена отвара.

Помогнете на горкия аптекар като напишете програма *kolbi*, която извежда номера на търсената колба.

Ограничения: $N \leq 1\,000\,000$

Вход: Броят на колбите и количеството на отварата. Отварата винаги се събира в поне една от колбите (т.е. винаги има решение).

Изход: Номерът на най-малката колба, която побира отварата

Вход:

10 5

Изход:

5

Обяснение на примера: Най-малката колба, която можем да ползваме, е 4, но тя е с четен номер и затова избираме следващата по-големина.

ЗАКЛЮЧЕНИЕ

Състезателна група Е (IV – V клас) започва официалното си съществуване от 2004 година. Част от колегията за извънкласна работа по информатика работи за обогатяване на литературния фонд за тази група. Липсата на задачи винаги е стояла като проблем, както и разработването на методика. Към този момент се разполага с голямо разнообразие от състезателни задачи, но все още има нужда от опорни такива. Тази тема допълва примерната учебна програма, по която се работи от 2010 година. Тя бе предизвикана от интереса на учениците в частта „Оператори за цикъл“.

След направено проучване сред ученици могат да се направят следните изводи.

Задачите са трудни за третокласниците. Четвъртокласниците се справят самостоятелно със задачите от първия тип, втори тип – първа задача, и четвърти тип. Останалите задачи решават с помощта на учителя. Петокласниците решават всички задачи. Помощ от учителя им е необходима за задача втора от третия тип.

Представената система от задачи е градивна, задачите следват дидактическите принципи за съзнателност (отделят се и се обобщават съществените свойства на понятията, учениците се приучават към рационални прийоми за работа над учебния материал), за достъпност – от лекото към трудното, от известно към неизвестно, за трайност на знанията и уменията и индивидуален подход.

БЕЛЕЖКИ

1. Infos (2014). <http://www.math.bas.bg/infos/files/2009-11-10-u4ebnaprograma> - активен 16.03.2014.

ЛИТЕРАТУРА

Андреев, М. (2003). *Процесът на обучението. Дидактика*. София: Университетско издателство „Св. Климент Охридски“.

- Вишин, Я. (1985). *Методика за решаване на математически задачи*. София: Народна просвета.
- Гъров, К. (2008). *Теория и практика на подготовката на изяви и таланти-ви ученици за участие в олимпиади и състезания по информатика и информационни технологии*. София: Докторска дисертация по научна специалност 05.07.03.
- Дурева-Тупарова, Д. (2003), *Проблеми от методиката на обучение по информатика и информационни технологии*. Благоевград: Университетско издателство при Югозападен университет „Неофит Рилски“.
- Келеведжиев, Е. & Дженкова, З. (2004). *Алгоритми, програми и задачи, ръководство за начална подготовка по информатика за олимпиади и състезания*. София: Регалия 6.
- Йовчева, Б. & Иванова, И. (2006). *Първи стъпки в програмирането на C/C++*. Варна: КЛМН.
- Христова, П. & Григорова, К. (2011). *Извънкласна работа по информатика, ръководство за упражнения – 1 част*. Русе: Университетско издателство.

PROGRAMMING OF NUMBER SERIES

Abstract. A system of supporting tasks related to the application of number series from Mathematics in Informatics is developed. Number series are used to form knowledge and skills of students for cyclic algorithms. The article is focused on competition group E in Informatics (10 – 11 year old students).

✉ **Dr. Ivaylo Staribratov**

High School of Mathematics „Acad. Kiril Popov“
11, Chemshir Str.
Plovdiv
E-mail: ivostar@abv.bg

✉ **Ms. Tsvetana Dimitrova**

High School of Mathematics „Acad. Kiril Popov“
11, Chemshir Str.
Plovdiv
E-mail: tsetsi.dimitrova.2010@gmail.com