

No

Решение:

Решението изисква познаване на отделянето на елементи от текст, сравнение на многоцифрени числа и признак за делимост на 3. Програмата съдържа: функция за сравнение на многоцифрени числа; функция за отделяне на цифрите на число от текст и запомнянето им в масив с подравняване към определена дължина, едновременно с това се намира и сумата от цифрите на числото. Използвайки тези функции обхождаме целия текст отделяме число от текста, проверяваме дали е кратно на 3 и ако да го сравняваме с намереното до момента максимално число. Накрая се извежда най-голямото число кратно на 3.

```
#include<iostream.h>
#define d 80
int a[d],b[d],n,m;
char ch;
void sr(int x[],int y[],int z[])
{int k,i=0;
  while (i<d&&x[i]==y[i]){z[i]=x[i];i++;}
  if (x[i]>y[i])for(k=i;k<d;k++) z[k]=x[k];
  else for(k=i;k<d;k++) z[k]=y[k];
  //for (k=0;k<d;k++)cout<<z[k];
}

int st(int x[])
{int k,i=-1,br=0;
  while (ch>='0'&&ch<='9')
  {i++;
   x[i]=ch-'0';
   br+=x[i]; cin.get(ch);
  }
  for (k=i;k>=0;k--) x[d-1-i+k]=x[k];
  for (k=0;k<d-1-i;k++) x[k]=0;
  return br;}

void main()
{
  a[0]=-1;for(m=1;m<d;m++) a[m]=0;
  do
  {cin.get(ch);
   if (ch>='0'&&ch<='9') if (st(b)%3==0) sr(a,b,a) ;
  }
  while (ch!='#');
  if (a[0]==-1) cout<<"No";
  else { n=0; while (n<d&&a[n]==0) n++;
        if (n==d) cout<<0; else for (m=n;m<d;m++)cout<<a[m];
      }
  cout<<"\n";
}
```

```
}
```

Решение на задачата на Pascal

```
program aa;
const d=80;
type mas=array[1..d]of integer;
var a,b:mas;n,m,i:integer;ch:char;
procedure sr(x,y:mas;var z:mas);
var i:byte;
begin i:=1;
  while (i<=d) and (x[i]=y[i]) do
    i:=i+1;
  if x[i]>y[i] then z:=x
  else z:=y;
end;

procedure st(var x:mas;var br:integer);
var k,i:integer;
begin
  i:=0;br:=0;
  while ch in ['0'..'9'] do
    begin
      i:=i+1;
      x[i]:=ord(ch)-ord('0');
      br:=br+x[i]; read(ch);
    end;
  for k:=i downto 1 do x[d-i+k]:=x[k];
  for k:=1 to d-i do x[k]:=0;
end;

begin
  a[1]:=-1;for m:=2 to d do a[m]:=0;
  repeat
    read(ch);
    if ch in ['0'..'9'] then begin
      st(b,i);
      if i mod 3=0 then sr(a,b,a)
      end
  until ch='#';
  if (a[1]=-1) then write('No') else
  begin
    n:=1; while (n<=d)and(a[n]=0) do n:=n+1;
    if n>d then write(0) else for m:=n to d do write(a[m]);
  end;
  writeln;
```

end.

Ето още едно решение, предложено от Бисерка Йовчева:

Решението е подобно на авторовото, но използва различни средства за програмиране, предимно възможностите на стандартната библиотека на езика C++, известна като STL. Тук за записа на дългите числа ще използвам вектори с цели елементи. Функцията **numcmp** приема като параметри два вектора и тяхната дължина и връща като резултат 1 ако в първия е записано число по-малко от числото, записано във втория, -1 ако числото във втория вектор е по-малко и 0 ако двете числа са равни.

```
#include<iostream>
#include<vector>
#include<string>
using namespace std;
int numcmp(vector<int> a, vector<int> b, int n)
{
    int i;
    for (i=0; i<n; i++)
        if (a[i]<b[i]) return 1;
        else
            if (a[i]>b[i]) return -1;
    return 0;
}
void vector_print(vector<int> a)
{
    int i, n=a.size();
    for (i=0; i<n; i++)
        cout<<a[i];
    cout<<endl;
}
int main()
{
    vector<int> a, m;
    int s=0, l=0, num, mn=1;
    char ch;
    m.push_back(0);
    do
    {
        cin.get(ch);
        num=ch-'0';
        if (num>9 || num<0)
        {
            if (s)
            {
                if (s%3==0)
                {
                    l=1;
                    int k= a.size();
                    if (k>mn || k==mn && (numcmp(m, a, k)==1))
                        {mn=k, m=a;}
                }
            }
        }
    }
```

```

        s=0;a.clear();
    }
}
else
{
    a.push_back(num);
    s+=num;
}

}
while(ch!='#');
if(l)vector_print(m);else cout<<"No\n";
return 0;
}

```

Задача D2. Влакове

В село Клавиатурмишково са открити находища на нефт и въглища в толкова големи количества, че за да се извозят е нужно да се създаде отделна ж.п. гара.

Г-н Мишо Клавишов, кметът на село Клавиатурмишково е изправен пред сериозен проблем! Нужна му е компютърна програма TRAINS.EXE, която чертае схема на гарата, заедно с влаковете, които в даден момент са спрели на гарата.

На ж.п. гарата има N ($1 \leq N \leq 9$) коловоза, всеки от които дълъг по M ($1 \leq M \leq 50$) единици, като единица ж.п. линия на схемата се отбелязва със символа „#”. Всеки от влаковете има точно по един локомотив, който на схемата се отбелязва с “(Н Н)”. Всеки влак може да има два вида вагони – за въглища (отбелязва се с „[XXX]”) и вагон-цистерна (отбелязва се с “(О О)”). Вагоните с въглища се композират винаги преди вагоните-цистерни. Броят на вагоните е така зададен, че целият влак да може да се визуализира на схемата.

Входни данни

На първия ред от клавиатурата се въвеждат 2 цели числа – N и M . На всеки от следващите N реда има по три цели числа – P , C , F .

P е позицията на локомотива в брой единици от началото на коловоза. Ако $P=0$, то коловозът е празен и стойностите на C и F за този коловоз нямат значение.

C е броят на вагоните с въглища.

F е броят на вагоните-цистерни.

Изходни данни

На екрана се извежда схема на гарата, заедно с влаковете, като всеки коловоз се показва на нов ред.

Пример:

В примера по-долу правим схема на гара с 3 коловоза, всеки от които с дължина 32 единици. На 1-ви коловоз имаме влак, който започва от 4-та позиция и има 1 вагон с въглища и 2 вагона-цистерни. 2-ри коловоз е празен. На 3-ти коловоз има влак, който започва от 3-та позиция и има 5 вагона-цистерни.

Примерен вход:

```

3 32
4 1 2
0 2 3

```

3 0 5

Примерен изход:

```
###(H H)[XXX](O O)(O O)#####  
#####  
##(H H)(O O)(O O)(O O)(O O)(O O)
```

Решение:

За решението на задачата ще използваме три едномерни масива, в които ще записваме съответно – в pos[i]- позицията на локомотива в i-тия коловоз, в c[i] – броя на вагоните с въглища, а в f[i]- броя на цистерните.

Единствената важна преценка е да се изчисли броят на символите, оставащи след последния вагон.

Те се изчисляват, като от дължината на коловоза М се извади:

$5 * (1 + F + C) + \text{позиция на локомотива}$, където F е броя на вагоните с въглища, а C- броя на цистерните.

Реализация на алгоритъма на езика C++

```
#include <cstdlib>  
#include <iostream>  
using namespace std;  
int main()  
{  
    int i,j,m,n;  
    int pos[9],c[9],f[9];  
    cin>>n>>m;  
    for (i=0;i<n;i++)  
        cin>>pos[i]>>c[i]>>f[i];  
    for (i=0;i<n;i++)  
    {  
        if (pos[i]>0)  
        {  
            for (j=1;j<pos[i];j++) cout<<'#';  
            cout<<"(H H)";  
            for (j=0;j<c[i];j++) cout<<"[XXX]";  
            for (j=0;j<f[i];j++) cout<<"(O O)";  
            for (j=5*(1+c[i]+f[i])+pos[i];j<=m;j++) cout<<'#';  
        }  
        else  
            for (j=0;j<m;j++) cout<<'#';  
        cout<<'\\n';  
    }  
}
```

Задача D3. Библиотека

Първата задача на библиотекарката Книжка Книгова на новото и работно място не била от най – любимите и. Тя трябвало да подреди N книги на шкаф с много рафтове, така че след като някой поиска дадена книга да може да я намира за най – малко време.

Ако подреди всичките N книги само на един рафт, времето необходимо за намирането, на коя да е книга ще е N секунди, а ако раздели книгите на K рафта, то времето, необходимо за намирането на книга от кой да е рафт ще е $K + P$ секунди, където P е броя на книгите на съответния рафт (т.е. K секунди, за да намери рафта и още P секунди, за да намери търсената книга). Например, ако има 7 книги, Книжка може да сложи всички книги на един рафт и тогава ще и отнеме 7 секунди за да намери коя да е от тях. Ако тя раздели 7 книги на 2 рафта с по 3 и 4 книги всеки, то ще и отнеме първо 2 секунди за да намери правилния рафт и след това още 3 или 4 секунди за намиране на всяка една от книгите, но след като незнаем коя книга ще търсим общото време за търсене ще бъде 6 секунди. Помогнете на библиотекарката да разбере колко ще е минималното време, необходимо намиране на произволна книга.

Напишете програма **MINITIME.EXE**, която от стандартния вход чете едно цяло число $N(1 \leq N \leq 107)$ – броя на книгите, които Книжка трябвало да подреди и отпечатва на стандартния изход едно число – минималното време, за което тя ще може да намери коя да е от книгите.

Примерен вход:

7

Примерен изход:

6

Примерен вход:

3

Примерен изход:

3

Решение:

Очевидно е, че трябва да намираме минималното време разпределяйки равномерно книгите по рафтове, започвайки от разпределение на книгите на един рафт, след това на два рафта и така разпределяйки ги до $N/2$ рафта включително. Тук интересно е смятането на времето за всяко едно разпределение по рафтове. Ако разделим N книги на K рафта то имаме две възможности. Първата е ако книгите се разпределят по равно на всичките K рафта т.е. K дели N без остатък. Тогава минималното време ще е K секунди (броя на рафтовете) и още N/K секунди – броя на книгите на един рафт. Но ако книгите не могат да се разделят по равно на всичките K рафта, тогава трябва да ги разпределим на $K+1$ рафта, където първите K рафта ще са с по равен брой книги, а $K+1$ -вия ще е с толкова книги колкото са останали след разполагането на първите K рафта т.е. с $N \% K$ книги. Тогава минималното време ще е $K+1$ секунди (за броя на рафтовете) и още N/K секунди за броя на книгите, защото на $K+1$ -вия рафт броя на книгите ще е винаги по – малък от колкото в горните рафтове.

Реализация на алгоритъма на езика C++

```
#include <iostream>
using namespace std;
int main()
{
    int n,mint,i,p,t;
    cin>>n;
    mint=n;
```

```
for (i=2; i<=n/2; i++)
{
    if (n%i)p=1;
    else p=0;
    t=(n/i)+(i+p);
    if (t<mint) mint=t;
}
cout<<mint<<endl;
return 0;
}
```