

Estruturas de Controle de Fluxo

Com base no que foi visto, resolva o exercício a seguir:

Construa um programa que receba uma sequência de números inteiros, positivos de três dígitos, e gere como saída, para cada valor pertencente à sequência, outro número formado pelos dígitos invertidos do número lido. O final da sequência é identificado pelo valor zero.

Exemplo de entrada:

123

901

530

0

Exemplo de saída:

321

109

035

Dica: Observe os resultados das funções Quociente e Resto de um número por 10.

```
#include <stdio.h>
int main()
{
    int N;
    do
    {
        scanf("%d", &N);
        if (N!=0)
            printf ("%03d\n", (N%10*100) + (N/10%10*10) + (N/10/10));
    }while(N);
}
```

```
#include <stdio.h>
int main()
{
    int N;
    do
    {
        scanf("%d", &N);
        if (!N )
            break;
        printf ("%03d\n", (N%10*100) + (N/10%10*10) + (N/10/10));
    }while(1);
}
```

Estruturas de Controle de Fluxo

Exercício 8:

Com base no que foi visto, resolva o problema intitulado “Jogo de Varetas” aplicado na primeira fase da Maratona de Programação no ano de 2007.

Há muitos jogos divertidos que usam pequenas varetas coloridas. A variante usada neste problema envolve a construção de retângulos. O jogo consiste em, dado um conjunto de varetas de comprimentos variados, desenhar retângulos no chão, utilizando as varetas como lados dos retângulos, sendo que cada vareta pode ser utilizada em apenas um retângulo, e cada lado de um retângulo é formado por uma única vareta. Nesse jogo, duas crianças recebem dois conjuntos iguais de varetas. Ganha o jogo a criança que desenhar o maior número de retângulos com o conjunto de varetas.

Dado um conjunto de varetas de comprimentos inteiros, você deve escrever um programa para determinar o maior número de retângulos que é possível desenhar.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N que indica o número de diferentes comprimentos de varetas ($1 \leq N \leq 1.000$) no conjunto. Cada uma das N linhas seguintes contém dois números inteiros C_i e V_i , representando respectivamente um comprimento ($1 \leq C_i \leq 10.000$) e o número de varetas com esse comprimento ($1 \leq V_i \leq 1.000$). Cada comprimento de vareta aparece no máximo uma vez em um conjunto de teste (ou seja, os valores C_i são distintos). O final da entrada é indicado por $N = 0$.

Saída

Para cada caso de teste da entrada seu programa deve produzir uma única linha na saída, contendo um número inteiro, indicando o número máximo de retângulos que podem ser formados com o conjunto de varetas dado.

Exemplo de Entrada

1
10 7

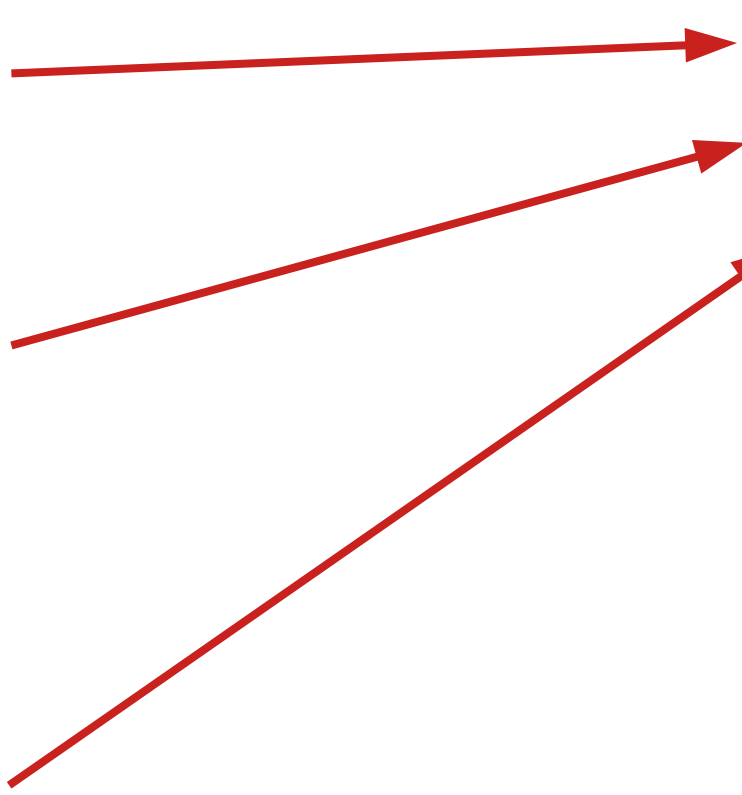
4
50 2
40 2
30 4
60 4

5
15 3
6 3
12 3
70 5
71 1

0

Exemplo de Saída

1
3
2



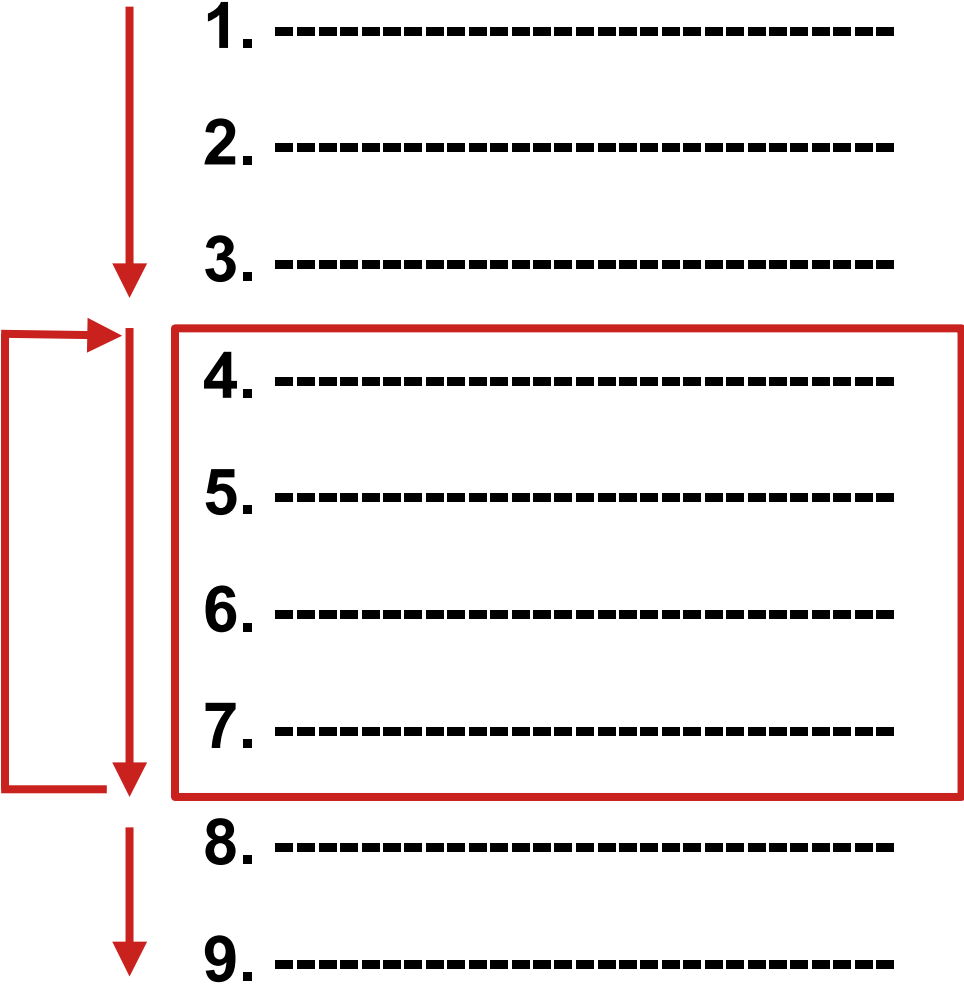
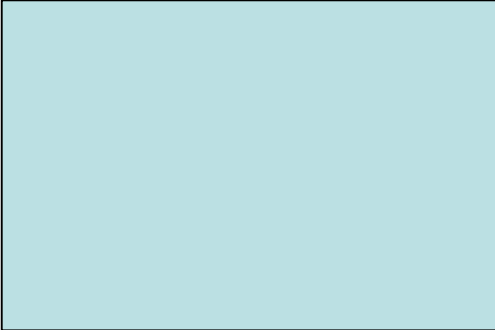
```
#include <stdio.h>
int main()
{
int N, Ci, Vi, i, numRetangulos;
do {
scanf("%d", &N);
if (N==0)
break;
numRetangulos=0;
i=0;
while (i<N) {
scanf("%d %d", &Ci, &Vi);
numRetangulos += Vi/2;
i++;
}
printf ("%d\n", numRetangulos/2);
}while(1);
}
```



Estruturas de Controle de Fluxo

Laços de Repetição (for)

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

3. Laços de repetição

...

```
for (<instruçãoX>; <expressão_lógica>; <instruçãoY>)
```

```
{
```

```
    <instruçãoZ>
```

```
    ...
```

```
}
```

...

Estruturas de Controle de Fluxo

3. Laços de repetição

```
for (i=0,j=0;i<10&&j<20;i++,j+=2)  
{  
    . . .  
}
```

```
for (i=0;i<1000;i++);  
printf (“\nEscreve qualquer coisa!\n”);  
/*O printf anterior não está contido no for!*/
```

Estruturas de Controle de Fluxo

3. Laços de repetição

Exemplo:

O programa a seguir recebe um número natural e retorna o seu fatorial na saída padrão.

```
#include <stdio.h>
int main()
{
int i,num,fat=1; long int i,num,fat=1;
do
{
    printf ("Digite um numero natural:");
    scanf ("%ld",&num);
}while(num<0);
for (i=2;i<=num;i++) for (i=2; i<=num; fat*=i, i++);
fat*=i; for (i=2; i<=num; fat*=i++);
printf("\n0 fatorial do numero%ldeh:%ld\n", num, fat);
}
```

$$n! = n * (n-1) * (n-2) * \dots * 1$$

$$4! = 4 * 3 * 2 * 1 = 24$$

Estruturas de Controle de Fluxo

3. Laços de repetição

Exercício 9:

Construa um programa que leia da entrada padrão uma sequência de números inteiros positivos, onde o final da sequência é identificado pelo valor zero. O programa deve determinar se os números recebidos são ou não quadrados perfeitos.

Obs.1: As entradas devem ser validadas.

Obs.2: Um número é um quadrado perfeito quando tem um número inteiro como raiz quadrada.

```
#include <stdio.h>
int main() {
    int num, c, p;
    do {
        do {
            printf ("Entre com o valor natural: ");
            scanf ("%d",&num);
        }while(num<0);
        if (num) {
            c=1;
            p=c*c;
            while (p<num) {
                c++;
                p=c*c;
            }
            if (p==num)
                printf("\n%d eh quadrado perfeito", num);
            else
                printf("\n%d nao eh quadrado perfeito", num);
        }
    }while(num);
}
```

```
#include <stdio.h>
int main() {
    int num, c, p;
    do
    {
        do
        {
            printf ("Entre com um valor natural (zero para finalizar): ");
            scanf ("%d", &num);
        } while (num < 0);
        if (num)
        {
            for (c=1, p=c*c; p<num; p=++c*c);
            if (p==num)
                printf ("\n%d eh quadrado perfeito", num);
            else
                printf ("\n%d nao eh quadrado perfeito", num);
        }
    } while (num);
}
```



```
#include <stdio.h>
#include <math.h>
int main() {
    int num;
    do {
        do {
            printf ("Entre com um valor natural (zero para finalizar): ");
            scanf ("%d",&num);
        }while(num<0);
        if (num)
            if (pow((int)sqrt(num),2)==num)
                printf("\n%d eh quadrado perfeito.\n", num);
            else
                printf("\n%d nao eh quadrado perfeito.\n", num);
        }while(num);
    }
}
```



Estruturas de Controle de Fluxo

Exercícios – Parte 1

Estruturas de Controle de Fluxo

3. Laços de repetição

Exercício 10:

Construa um programa que calcule o M.M.C. entre dois números naturais lidos.