

1.

/* Este é um exercício clássico. A ideia central é usar um acumulador, mas em vez de ir somando valores nele, vamos multiplicando. */

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

```
int main ()
```

```
{
```

```
    int n;
```

```
    int fatorial = 1;
```

```
    scanf ("%d", &n);
```

```
    while (n > 1) // Vamos com n, n-1, n-2, n-3... até 2.
```

```
    {
```

```
        fatorial *= n;
```

```
        n--;
```

```
    }
```

```
    printf ("%d\n", fatorial);
```

```
    return (0);
```

```
}
```

2.

```
/* Para resolver esta questão, precisamos sempre guardar 2 números: o anterior e o atual. Comparando os 2 números, sabemos se a sequência é crescente ou decrescente - o atual precisa ser maior que o anterior, ou vice-versa. */
```

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

```
int main ()
```

```
{  
    int i, n, anterior, atual;  
    int crescente; // Flag que diz se a sequência é crescente.  
    int decrescente; // Flag que diz se a sequência é decrescente.
```

```
    // Lê o n e o primeiro valor.
```

```
    scanf ("%d", &n);
```

```
    scanf ("%d", &anterior);
```

```
    /* Começamos supondo que a sequência é crescente E decrescente. (!?)
```

```
    Se não for um dos casos, mudamos a flag. */
```

```
    crescente = 1;
```

```
    decrescente = 1;
```

```
    i = 1; // A contagem começa em 1 porque já lemos o 1o valor!
```

```
    while (i < n)
```

```
    {  
        scanf ("%d", &atual);
```

```
        if (atual <= anterior)
```

```
            crescente = 0;
```

```
        if (atual >= anterior)
```

```
            decrescente = 0;
```

```
        // O atual desta iteração vira o anterior da próxima iteração.
```

```
        anterior = atual;
```

```
        i++;
```

```
    }
```

```
    if (crescente)
```

```
        printf ("Crescente!\n");
```

```
    else if (decrescente)
```

```
        printf ("Decrescente!\n");
```

```
    else
```

```
        printf ("Nem um nem outro!\n");
```

```
    return (0);
```

```
}
```

3.

```
#include <stdio.h>

int main ()
{
    int quociente, resto;
    int numerador, denominador;

    scanf ("%d/%d", &numerador, &denominador);

    /* A cada iteração, subtrai o denominador do numerador e aumenta o quociente
       em 1. Como estamos mexendo no numerador, no final do loop o valor não vai
       mais ser o digitado, e sim o resto da divisão. */
    for (quociente = 0; numerador >= denominador; quociente++)
        numerador -= denominador;

    resto = numerador;

    printf ("Quociente: %d, Resto: %d\n", quociente, resto);

    return (0);
}
```

4.

```
/* O "truque" aqui é ir olhando sempre a casa das unidades (resto da divisão por
   10), e dividindo o número por 10 sucessivamente, até que ele chegue a 0. */

#include <stdio.h>

int main ()
{
    int n, digito;
    int achou = 0;

    scanf ("%d", &n);
    scanf ("%d", &digito);

    // Vamos dividir o número por 10 até que ele "suma".
    while (n > 0 && !achou)
    {
        // Verifica se a unidade é o dígito procurado.
        if (n % 10 == digito)
            achou = 1;

        n /= 10;
    }

    if (achou)
        printf ("Tem um %d!\n", digito);
    else
        printf ("Nao tem um %d!\n", digito);

    return (0);
}
```

5.

```
#include <stdio.h>

int main ()
{
    int n;
    int n_anterior = 0;
    int tam_sequencia = 0; // O tamanho da sequência crescente atual.
    int maior_sequencia = 0; // O tamanho da maior sequência até agora.

    /* Começamos com tudo em 0, de forma que o primeiro número é maior que o
       n_anterior e a primeira sequência tem tamanho pelo menos igual a 1. */

    scanf ("%d", &n);

    while (n > 0)
    {
        if (n > n_anterior)
        {
            tam_sequencia++;
            if (tam_sequencia > maior_sequencia)
                maior_sequencia = tam_sequencia;
        }
        else
            tam_sequencia = 1; // Ops, quebrou a sequência, vamos recomeçar.

        n_anterior = n; // Lembra do n anterior.
        scanf ("%d", &n); // Lê o próximo número.
    }

    printf ("%d\n", maior_sequencia);

    return (0);
}
```