

**1. Escreva um aplicativo em C mostra todos os números ímpares de 1 até 100.**

Essa é bem simples. Basta percorrermos os números de 1 até 100, e fazer um teste condicional IF dentro do laço WHILE, para exibir somente aqueles números que são ímpares, ou seja, aqueles números que deixam resto 1 quando divididos por

**2. Escreva um aplicativo em C que mostra todos os números pares de 1 até 100.**

Utilize o exercício anterior, basta você alterar apenas um número.

**3. Escreva um aplicativo em C que recebe inteiro e mostra os números pares e ímpares (separados, em duas colunas), de 1 até esse inteiro.**

Como o primeiro número é ímpar (1), os ímpares serão exibidos primeiro. Após cada ímpar damos o espaço de um TAB (t), e na mesma linha imprimimos o par, e logo em seguinte o caractere new line (\n).

**4. Escreva um programa que pergunte ao usuário quantos alunos tem na sala dele. Em seguida, através de um laço while, pede ao usuário para que entre com as notas de todos os alunos da sala, um por vez. Por fim, o programa deve mostrar a média, aritmética, da turma.**

Para resolver este exercício, vamos precisar de várias variáveis de apoio.

A variável 'total' vai receber o número total de alunos em uma sala. A variável 'count' será a que vai mudar dentro do laço WHILE. E ela começa do primeiro aluno e termina no último. A cada iteração do laço while, pedimos uma nota, que é somada a variável do tipo float 'soma'. A média aritmética é dada por essa soma (de todas as notas), dividida pelo número de alunos na sala.

**5. Escreva um programa em C que solicita 10 números ao usuário, através de um laço while, e ao final mostre qual destes números é o maior.**

Vamos usar a nossa velha e conhecida variável 'count' pra contar de 1 até 10 no laço while.

Inicialmente, também, criamos a variável 'maior', que irá armazenar o maior número. Inicializamos ela com valor 0.

A cada iteração do laço while pedimos um número ao usuário, e armazenamos sempre na variável 'num'.

Em seguida fazemos uma comparação com o teste condicional IF ELSE para saber se esse número digitado é maior que o 'maior'. Se for, é porque esse número é o maior que já foi digitado, então fazemos com a variável 'maior' receba o valor desse 'num'. Caso o 'num' não seja maior que 'maior', nada acontece e 'maior' continua armazenando o maior número.

**6. Escreva um programa em C que solicita 10 números ao usuário, através de um laço while, e ao final mostre os dois maiores números digitados pelo usuário.**

Os dois primeiros números são armazenados fora do laço while.

Em seguida, é feita uma lógica simples entre esses dois para saber qual vai ser o maior e qual vai ser o segundo maior.

Agora que já temos os números certos nas variáveis 'maior' e 'segundo\_maior', vamos ao laço WHILE que vai pedir do número 3 até o número 10.

Após o número ser inserido, ele vai ser armazenado na variável 'num'.

Agora vamos aos testes:

1. Primeiro checamos se 'num' é maior que 'maior', se for, é porque esse número será o novo número maior, e o antigo maior número temos que colocar na variável 'segundo\_maior'.

2. Caso não seja maior que 'maior', pode ser que ele seja maior que 'segundo\_maior' e fazemos o teste.

Se for, somente o 'segundo\_maior' muda de valor.

E se 'num' não for maior que 'maior' nem 'segundo\_maior', nenhuma alteração é feita.

7. **Escreva um programa que lê o tamanho do lado de um quadrado e imprime um quadrado daquele tamanho com asteriscos. Seu programa deve funcionar para quadrados com lados de todos os tamanhos entre 1 e 20.**

Por exemplo, para lado igual a 5:

```
*****
*****
*****
*****
*****
```

Note que sempre existirão 'lado \* lado' asteriscos. Vamos usar o while para imprimir todos os 'lado\*lado' asteriscos.

Note também que a cada 'lado' asteriscos, ocorre uma quebra da linha.

No exemplo do lado=5: quando chegamos no asterisco de número 5, não imprimimos somente '\*', mas '\*\n'

O mesmo para o asterisco de número 10, pro de número 15, ...e pro de número 20

Ou seja, após o asterisco múltiplo de 5 ser impresso na tela, devemos colocar uma quebra da linha.

Como chamamos números que são múltiplos de 'lado'? Ora, pelo módulo (ou resto da divisão).

8. **Escreva um programa que lê o tamanho do lado de um quadrado e imprime um quadrado daquele tamanho com asteriscos e espaços em branco. Seu programa deve funcionar para quadrados com lados de todos os tamanhos entre 1 e 20.**

**Para lado igual a 5:**

```
* * * * *
*       *
*       *
*       *
*       *
* * * * *
```

Primeiro vamos imprimir a primeira linha, que tem 'lado' asteriscos com um while.

Fazemos o mesmo para a última linha.

Agora vamos imprimir as linhas que não são completamente preenchidas de asteriscos, as que tem espaços em branco.

Note que essas linhas possuem um asterisco na primeira e na última posição.

Antes nós imprimíamos lado\*lado caracteres. Agora, como tiramos duas linhas, vamos imprimir só 'lado\*(lado-2)' linhas.

Se o caractere que formos imprimir for o primeiro da fila, imprimimos '\*'.

Para sabermos isso, basta fazer um teste para ver se o contador (que vai de 1 até lado\*(lado-2)) deixa resto 1 quando dividido por 'lado'.

Se for o último asterisco da linha, devemos imprimir '\*\n', pois irá pular de linha.

Para detectar esse último asterisco, basta ver que é o caractere que, quando dividido por 'lado', deixa resto 0.

Se não for o primeiro nem o último caractere, imprimimos um espaço em branco: ' '

9. **Escreva um programa que pergunta um número ao usuário, e mostra sua tabuada completa (de 1 até 10).**

**Primeiro pedimos um número ao usuário, e fazemos com que esse número seja multiplicado por 1, por 2, ...até por 10.**

Esse número que muda (1, 2, 3, ...,10), será gerado pelo laço while e uma variável de apoio 'count':

Se você lembrar bem, quando estudamos as variáveis do tipo caractere, char, dissemos que, na verdade, ela eram representadas por inteiros de 0 até 255.

Mostre a tabela completa do código ASCII.

Basta usarmos o laço while e uma variável de apoio para imprimirmos todos os números, de 0 até 255.

Mas ao invés de mostrar o número com o '%d', vamos usar '%c', pois assim iremos mostrar o caractere correspondente a cada número desses.