

Nome: \_\_\_\_\_

RA: \_\_\_\_\_

Assinatura: \_\_\_\_\_

### Instruções

- Desligue seu celular e guarde-o na mochila!
- É permitido consultar material individual em papel ou digital. No caso de material digital, este não poderá ser aberto via *browser*.
- A única página que poderá ser aberta via *browser* é a página do Susy.
- Tentativa de fraude/cola resultará em 0 na prova para todos os envolvidos.
- Nenhum aluno deve deixar a prova antes de 1h00 e ao sair deve deixar o enunciado da prova (com nome, RA e assinatura).
- **Compile e teste seus programas antes de submetê-los ao Susy!**

### Questão 1:

O programa abaixo imprime as  $n$  primeiras potências para os inteiros positivos de 2 a  $z$ , sendo  $n$  e  $z$  entradas fornecidas pelo usuário. Entretanto, o código está incompleto. Sua tarefa consiste em completar o programa implementando uma função que compute a potência  $x^y$  para valores  $x$  e  $y$  passados como parâmetros (não use a biblioteca `math.h`). Sua função deve ter o seguinte protótipo:

**long pot (int x, int y);**

```
#include <stdio.h>

/*inclua aqui a implementacao da funcao pot */

int main () {
    int z, n, i, j;
    long potencia;

    scanf("%d", &z);
    scanf("%d", &n);
    if ((z < 2) || (n < 1))
        printf("Entrada invalida!\n");
    else {
        for (i = 2; i <= z; i++) {
            if (n==1)
                printf("primeira potencia de %d: ", i);
            else
                printf("%d primeiras potencias de %d: ", n, i);
            for (j = 0; j < n; j++) {

                /*inclua aqui a chamada para a funcao pot */

                printf("%ld ", potencia);
            }
            printf("\n");
        }
    }
    return 0;
}
```

**Entrada:**

Um inteiro  $z \geq 2$  e um inteiro  $n \geq 1$  tal que sejam calculadas as  $n$  primeiras potências para cada inteiro no intervalo  $[2, z]$ .

**Saída:**

Para cada inteiro  $i$  no intervalo  $[2, z]$ , se  $n > 1$ , deve-se imprimir:

$n$  primeiras potências de  $i$ :  $p_1 p_2 \dots p_n$

se  $n = 1$ , deve-se imprimir:

primeira potência de  $i$ :  $p_1$

onde  $n$  e  $i$  devem ser substituídos pelos respectivos valores e  $p_1 p_2 \dots p_n$  devem ser substituídos pelas respectivas potências.

Se  $n < 1$  e/ou  $z < 2$  a saída deve ser:

Entrada invalida!

**Exemplo de execução 1:**

4

8

8 primeiras potencias de 2: 1 2 4 8 16 32 64 128

8 primeiras potencias de 3: 1 3 9 27 81 243 729 2187

8 primeiras potencias de 4: 1 4 16 64 256 1024 4096 16384

**Exemplo de execução 2:**

7

3

3 primeiras potencias de 2: 1 2 4

3 primeiras potencias de 3: 1 3 9

3 primeiras potencias de 4: 1 4 16

3 primeiras potencias de 5: 1 5 25

3 primeiras potencias de 6: 1 6 36

3 primeiras potencias de 7: 1 7 49

**Exemplo de execução 3:**

3

1

primeira potencia de 2: 1

primeira potencia de 3: 1

**Exemplo de execução 4:**

3

0

Entrada invalida!

Nos exemplos acima as 2 primeiras linhas correspondem às entradas fornecidas pelo usuário e as linhas seguintes correspondem à saída fornecida pelo programa.

## Questão 2:

O programa abaixo recebe como entrada um inteiro positivo **n** e imprime como saída o produto dos dígitos decimais desse inteiro. Entretanto, o código está incompleto. Sua tarefa consiste em completar o código implementando uma função RECURSIVA que calcule o produto dos dígitos decimais de um inteiro positivo. Por exemplo, o produto dos dígitos de 154 é 20. Sua função deve ter o seguinte protótipo:

**int produto(int n)**

**Obs.: soluções iterativas receberão nota 0 mesmo que os resultados dos testes estejam corretos!**

```
#include <stdio.h>

/*inclua aqui a implementacao da funcao produto */

int main(){
    int n;
    scanf("%d",&n);
    if (n > 0)
        printf("Produto: %d\n", produto(n));
    return 0;
}
```

### Entrada:

Um inteiro positivo **n**.

### Saída:

Se a entrada for válida, seu programa deve imprimir como saída:

Produto:  $P_n$

onde  $P_n$  é o produto dos dígitos decimais de **n** calculado pela função **produto**.

### Exemplo de execução 1:

156  
Produto: 30

### Exemplo de execução 2:

208  
Produto: 0

### Exemplo de execução 3:

-89

Nos exemplos acima a primeira linha corresponde à entrada fornecida pelo usuário e a segunda linha corresponde à saída fornecida pelo programa.

### Observações gerais:

- O nome do arquivo submetido deve ser potencia.c para a questão 1 e produto.c para a questão 2
- O número máximo de submissões é 10.
- Para a realização dos testes automáticos, a compilação se dará da seguinte forma: gcc nomedoarquivo.c -o nomedoarquivo -Wall -Werror -ansi -pedantic.
- Não se esqueça de incluir no início do programa uma breve descrição dos objetivos, da entrada, da saída, seu nome e RA.

- Após cada submissão, você deve aguardar um minuto até poder submeter seu trabalho novamente.

**Critérios importantes:**

- Não se esqueça de indentar e comentar seu programa adequadamente.
- Não serão aceitas soluções contendo estruturas não vistas em sala (para este laboratório, poderão ser utilizadas estruturas condicionais, estruturas de repetição, operações aritméticas, e de entrada e saída e funções);
- O único header aceito para inclusão é **stdio.h**.