

Exercícios para Recuperação

1. Escreva um programa que faça a simplificação de frações. O usuário irá passar o numerador e denominador, e seu programa deverá mostrar como resultado a fração simplificada.

```

1 #include <stdio.h>
2
3 int MDC(int a, int b) {
4
5     int menor, i, mdc;
6
7     if (a < b)
8         menor = a;
9     else
10        menor = b;
11
12    for (i = 1; i <= menor; i++) {
13        if (a % i == 0 && b % i == 0)
14            mdc = i;
15    }
16    return mdc;
17 }
18
19 int main() {
20
21     int num, den, mdc;
22
23     printf("Digite o numerador e denominador, em sequencia:\n");
24     scanf("%d %d", &num, &den);
25
26
27     mdc = MDC(num, den);
28     num /= mdc;
29     den /= mdc;
30
31     printf("A fracao simplificada eh: %d/%d\n", num, den);
32
33     return 0;
34 }

```

2. Faça um programa para imprimir um triângulo de altura n e largura $2n - 1$. Por exemplo, para $n = 6$, a figura deve ficar como abaixo:

```

      *
     ***
    *****
   *********
  ***********
 *****

```

```

1 #include <stdio.h>
2
3 int main() {
4     int n, i, j, k, numAsteriscos, numEspacos;
5
6     printf("Digite o valor de N:\n");

```

```

7 scanf("%d", &n);
8
9 for (i = 1; i <= n; i++) {
10
11     numEspacos = n - i;
12     for (j = 0; j < numEspacos; j++) {
13         printf(" ");
14     }
15
16     numAsteriscos = 2*i - 1;
17     for (k = 0; k < numAsteriscos; k++) {
18         printf("*");
19     }
20
21     printf("\n");
22 }
23
24 return 0;
25 }
26

```

3. Um número forte (strong number) é um número no qual a soma do fatorial de cada um dos seus dígitos resulta no próprio número. Exemplo:

$$145 \rightarrow 1! + 4! + 5! = 1 + 24 + 120 = 145$$

Escreva um programa em C para verificar se o número dado é o número forte ou não.

```

1 #include <stdio.h>
2
3 int fatorial(int n) {
4     int fat = 1, i;
5
6     for (i = 1; i <= n; i++) {
7         fat *= i;
8     }
9
10    return fat;
11 }
12
13
14 int main() {
15     char num[5];
16     int somaFatorial = 0, i, digito;
17
18     printf("Digite um numero (maximo 5 digitos):\n");
19     scanf("%s", num);
20
21     for (i = 0; i < strlen(num); i++) {
22         // convertendo o digito para inteiro
23         digito = num[i] - '0';
24
25         somaFatorial += fatorial(digito);
26     }
27
28     if (somaFatorial == atoi(num)) {
29         printf("O numero eh forte\n");
30     } else {
31         printf("O numero NAO eh forte\n");
32     }
33
34     return 0;
35 }

```

4. Faça um programa em C para listar todos os números fortes de N dígitos, sendo N inteiro positivo dado pelo usuário.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 int fatorial(int n) {
6     int fat = 1, i;
7
8     for (i = 1; i <= n; i++) {
9         fat *= i;
10    }
11
12    return fat;
13 }
14
15
16 int verificaNumPerfeito(char *num, int casas) {
17     int somaFatorial = 0, i, digito;
18
19     for (i = 0; i < casas; i++) {
20         digito = num[i] - '0';
21         somaFatorial += fatorial(digito);
22     }
23     return somaFatorial == atoi(num);
24 }
25
26
27
28 void encontraNumerosPerfeitos(int casas) {
29     int i;
30     char *num;
31     int valorInicial = pow(10, casas - 1);
32     int valorFinal = pow(10, casas);
33
34     num = (char *)malloc(sizeof(char) * casas);
35
36     for (i = valorInicial; i < valorFinal; i++) {
37         //converte o numero (formato inteiro) para vetor de caracteres (string)
38         sprintf(num, "%d", i);
39
40         if (verificaNumPerfeito(num, casas))
41             printf("%d ", i);
42     }
43     printf("\n");
44 }
45
46
47 int main() {
48     int casas;
49
50     printf("Digite a quantidade de casas do numero\n");
51     scanf("%d", &casas);
52
53     encontraNumerosPerfeitos(casas);
54
55     return 0;
56 }
```

5. Faça um programa em C para receber o nome, o peso e a altura de cinco alunos do Coltec. A partir dos dados, devem-se fornecer as seguintes informações:

- **quem** é o mais alto (obviamente informando o nome)
- **quem** é o mais baixo (obviamente informando o nome)

- o peso médio do grupo.
- a altura média do grupo.

```

1  #include <stdio.h>
2
3  #define QUANTIDADE 5
4
5  int main() {
6      char nome[QUANTIDADE][50];
7      int peso[QUANTIDADE];
8      int altura[QUANTIDADE];
9      int i, iMaisAlto, iMaisBaixo, pesoMedio = 0;
10     int alturaMedia = 0;
11
12     for (i = 0; i < QUANTIDADE; i++) {
13         printf("Digite seu nome (maximo 50 caracteres)\n");
14         scanf("%s", nome[i]);
15
16         printf("Digite o seu peso\n");
17         scanf("%d", &peso[i]);
18
19         printf("Digite a sua altura (em cm)\n");
20         scanf("%d", &altura[i]);
21     }
22
23
24     iMaisAlto = 0;
25     iMaisBaixo = 0;
26     for (i = 0; i < QUANTIDADE; i++) {
27         pesoMedio += peso[i];
28         alturaMedia += altura[i];
29
30         if(altura[i] >= altura[iMaisAlto])
31             iMaisAlto = i;
32
33         if(altura[i] <= altura[iMaisBaixo])
34             iMaisBaixo = i;
35     }
36
37     pesoMedio /= QUANTIDADE;
38     alturaMedia /= QUANTIDADE;
39
40     printf("Aluno mais alto: %s\n", nome[iMaisAlto]);
41     printf("Aluno mais baixo: %s\n", nome[iMaisBaixo]);
42     printf("peso medio: %d\n", pesoMedio);
43     printf("altura media: %d\n", alturaMedia);
44
45     return 0;
46 }

```

6. Escreva um programa que leia um número inteiro positivo n , em seguida leia n números inteiros e imprima esses n números em ordem invertida. Exemplo de entrada:

```

n -> 4
numeros digitados: 10, 2, 33, 100
saida: 100, 33, 2, 10

```

O programa deve impor a restrição de n ser inteiro positivo.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {

```

```
5
6  int *numeros, quantidade, i;
7
8  printf("Digite a quantidade de numeros\n");
9  scanf("%d", &quantidade);
10
11
12  numeros = (int *) malloc(sizeof(int) * quantidade);
13
14  for (i = 0; i < quantidade; i++) {
15      printf("\nDigite um numero: ");
16      scanf("%d", &numeros[i]);
17  }
18
19
20  printf("\nNumeros digitados: ");
21  for (i = 0; i < quantidade; i++) {
22      printf("%d ", numeros[i]);
23  }
24
25
26  printf("\nSaida: ");
27  for (i = quantidade - 1; i >= 0; i--) {
28      printf("%d ", numeros[i]);
29  }
30 }
```