

INF 1620 – P1 - 11/04/03	Questão 1
Nome:	
Matrícula:	Turma

- (a) Escreva uma função em C que receba como parâmetro um número inteiro não negativo e retorne, como resultado da função, o valor do seu fatorial. Essa função deve obedecer o protótipo:

```
int fat (int n);
```

- (b) Dado um conjunto com n elementos, chama-se *arranjo simples* de taxa k , a todo agrupamento de k elementos distintos dispostos numa certa ordem. A fórmula para calcular o número total de arranjos de um conjunto de n elementos tomados k a k é dada por:

$$A_{n,k} = \frac{n!}{(n-k)!}$$

Usando a função fatorial do item (a), escreva um programa que capture via teclado dois números inteiros, representando n e k , e imprima o valor do número total de arranjos, $A_{n,k}$. Pode-se assumir que o usuário sempre fornecerá valores válidos.

(a)

```
int fat (int n)
{
    int i;
    int f=1;
    for (i=1; i<=n; i++)
        f = f*i;
    return f;
}
```

(b)

```
#include <stdio.h>

int main (void)
{
    int n, k;
    int a;
    scanf("%d %d", &n, &k);
    a = fat(n) / fat(n-k);
    printf("numero de arranjos = %d\n",a);
    return 0;
}
```

INF 1620 – P1 - 11/04/03	Questão 2
Nome:	
Matrícula:	Turma

O cálculo da média ponderada de valores v_i com pesos de ponderação dados pelos respectivos w_i é feito pela seguinte fórmula:

$$m = \frac{\sum v_i w_i}{\sum w_i}$$

Escreva uma função que, dados o número de elementos e os vetores dos valores e dos respectivos pesos, retorne o valor da média ponderada. Essa função deve obedecer o protótipo:

```
float ponderada (int n, float* v, float* w);

float ponderada (int n, float* v, float* w)
{
    int i;
    float num = 0.0; /* numerador */
    float den = 0.0; /* denominador */

    for (i=0; i<n; i++)
    {
        num = num + v[i]*w[i];
        den = den + w[i];
    }
    return num/den;
}
```

INF 1620 – P1 - 11/04/03	Questão 3
Nome:	
Matrícula:	Turma

Considere uma função, `minmax`, para calcular os valores mínimo e máximo dentre os elementos de um vetor de número reais. O programa abaixo ilustra uma utilização desta função, imprimindo os valores `-2.3` e `7.8` que representam os valores mínimo e máximo do vetor em questão..

```
#include <stdio.>
int main (void) {
    float v[7] = {4.5, 7.8, 6.5, -2.3, 2.3, 5.6, -1.1};
    float min, max;
    /* calcula valores mínimo e máximo do vetor */
    minmax (7, v, &min, &max);
    printf("min = %f  max = %f\n", min, max);
    return 0;
}
```

Implemente a função `minmax` de tal forma que o exemplo acima funcione adequadamente. A solução deve servir para o cálculo de valores mínimo e máximo de qualquer vetor, não podendo ser particularizada para o programa acima.

```
void minmax (int n, float* v, float* pmin, float* pmax)
{
    int i;

    *pmin = v[0];
    *pmax = v[0];

    for (i=1; i<n; i++)
    {
        if (v[i] < *pmin)
            *pmin = v[i];
        if (v[i] > *pmax)
            *pmax = v[i];
    }
}
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1620 – P1 - 11/04/03	Questão 4
Nome:	
Matrícula:	Turma

Diz-se que uma cadeia de caracteres é um palíndromo se ela pode ser lida, indiferentemente, da esquerda para a direita ou vice-versa. Assim, a cadeia de caracteres “I was stressed desserts saw I” representa um palíndromo. Escreva uma função para verificar se uma dada cadeia de caracteres é ou não um palíndromo. A função deve retornar 1 se for palíndromo e 0 se não for. Essa função deve obedecer o protótipo:

```
int palindromo (char* s);

int palindromo (char* s)
{
    int i;
    int n = strlen(s);

    for (i=0; i<n/2; i++)
    {
        if (s[i] != s[n-i-1])
            return 0;
    }
    return 1;
}
```

Funções da biblioteca padrão de C que podem ser úteis:

```
int strlen (char* s);
char* strcpy (char* s1, char* s2);
char* strcat (char* s1, char* s2);
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1620 – P1 - 11/04/03	Questão 5
Nome:	
Matrícula:	Turma

Implemente uma função que receba uma string e um número inteiro n como parâmetros, e retorne uma nova string com os n primeiros caracteres da string passada como parâmetro. Por exemplo, recebendo como parâmetros a string “Estruturas” e o número 3, essa função retornaria uma nova string, alocada dinamicamente, contendo a sequência de caracteres “Est”. Essa função deve obedecer o protótipo:

```
char* prefixo (char* s, int n);
```

Obs.: A string passada como parâmetro não pode ser alterada. Assuma que n é sempre menor do que o comprimento da string passada.

```
char* prefixo (char* s, int n)
{
    int i;
    char* p;

    p = (char*) malloc((n+1)*sizeof(char));

    for (i=0; i<n; i++)
    {
        p[i] = s[i];
    }
    p[n] = '\0';

    return p;
}
```

RASCUNHO

Respostas nesta folha não serão consideradas.

Não separe as folhas deste caderno.