

1. Sequencias de escape e caracteres especiais

ASCII = American Standard Code for Information Interchange

notação C	\n	\t	EOF	\r	\0	\\	\?	\'	\"
significado	newline	tab	end of file	retorno de linha (carriage return)	nulo (null)	barra invertida (backslash)	question mark	aspas simples (single quote)	aspas duplas (double quote)
ASCII nº	10	9	26	13	0	92	63	96	34

```
#include <stdio.h>
// para visualizar o efeito dos caracteres especiais \n , \t e \r
int main(){
    char c;

    c = '\n';
    printf("\n \n = ASCII %d\n", c);
    printf("linha1\nlinha2\n");
    c = '\t';
    printf("\n \t = ASCII %d\n", c);
    printf("texto1 \t texto2\n");
    c = '\r';
    printf("\n \r = ASCII %d\n", c);
    printf("texto1 \r texto2\n");
    system("pause");
    return 0;
}
```

'A' é um caracter simples (numero 65 na tabela ASCII)
 "A" é um "string" (cordão de caracteres) com um único caracter: A \0

OBS.: printf("texto1 \r texto2\n");
 tem o mesmo efeito que c = '\r'; printf("texto1 %c texto2\n", c);

3. Trabalhando com strings

EXEMPLO 1. Lendo um cordão de caracteres e contando as letras e palavras

```
#include <stdio.h>

#define IN 1 /* dentro de uma palavra */
#define OUT 0 /* fora de uma palavra */

int main(){
    char c, s[80]; // pressupondo que uma linha tenha no maximo 80 caracteres
    int i, nc, nw, state;

    // procedimento recomendado para ler uma linha de caracteres e colocar a mesma num string
    i=0;
    while((s[i] = fgetc(stdin)) != '\n') i++;
    s[i] = '\0';

    printf("\n%s\n", s);

    // contando os caracteres que interessam
    nc = i--;
    printf("\nEssa linha contem %d caracteres\n", nc);

    // contando as palavras (Kernighan e Ritchie)
    nw = 0;
    state = OUT;
    for(i=0; i<nc; i++){c=s[i];
        if(c == ' ' || c == '\n' || c == '\t') state = OUT;
        else if (state == OUT){ state = IN;
            ++nw; }
    }
    printf("\nEssa linha contem %d palavras\n", nw);

    system("pause");
    return 0;
}
```

EXERCÍCIO 1. (K&R) Acrescente ou modifique o programa acima para que ele imprima uma palavra por linha.

- 3. Escopo das variáveis
- 4. constantes pré-definidas
- 5. procedimentos seguros de leitura de dados

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

int forma;

double area( double R){
    return(forma*M_PI*R*R);
}

int main(){
    double R, S;

    forma = 0;
    while(forma != 1 && forma != 2){
        printf("circulo (1) ou esfera (2) ?\n");
        scanf("%d", &forma);
        switch (forma) {
            case 1: printf("\n circulo\n");
                    break;
            case 2: printf("\n esfera\n");
                    break;
            default: printf("\n valor invalido : %d\n", forma);
                    break;
        }
    }
    R = -1;
    while(R < 0){ printf("\nRaio ?\n");
                 scanf("%f", &R);
                 if(R < 0)printf("\nValor invalido : %g !!!\n", R);
    }
    if(forma == 2)forma = 4;
    S = area(R);
    printf("\nArea (raio %g) = %g\n\n", R, S);

    system("pause");
    return 0;
}
```

EXEMPLO 2. Um programinha que calcula ou a área de um círculo de raio R (πR^2) ou a área de uma esfera de raio R ($4\pi R^2$).

EXERCÍCIO 2. Complete o programa abaixo, que transforma ângulos de graus para radianos, ou vice-versa. Os passos são:

- pedir ao usuário que entre com a opção desejada (rad para grau ou grau para rad);
- ler o ângulo. Se for em graus, ler o mesmo na forma **graus minutos segundos**
- transformar para a unidade desejada. Se for para graus, responder na forma **graus minutos segundos**
- o ângulo final, se pedido em graus, deve estar entre -180° e $+180^\circ$.

Complete as funções `graus_dec()`, `graus_sex()`, `grad_to_rad()`, e `rad_to_graus_dec()`.

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

// o array gmt[ ] guarda graus em gms[0], minutos em gms[1] e segundos em gms[2]

int gms[3];

double graus_dec(){
    // esta função retorna o valor do ângulo em graus decimais, a partir dos valores guardados em gms[ ]
}

void graus_sex( double graus_dec ){
    // preenche os valores de gms[0], gms[1] e gms[2] a partir do valor do ângulo (dado em graus decimais)
}
```

```

double grad_to_rad( double graus_dec ){
    // recebe o valor do ângulo em graus decimais e retorna o mesmo em radianos
}

double rad_to_graus_dec( double rad ){
    // recebe o valor do ângulo em radianos e retorna o mesmo em graus decimais
}

int main(){
    double angulo;
    int opt,sinal;

    opt = 0;
    while(opt != 1 && opt != 2){
        printf("rad->graus (1) ou graus->rad (2) ?\n");
        scanf("%d", &opt);
        switch (opt) {
            case 1: printf("\n Entre com o angulo em radianos\n");
                scanf("%f", &angulo);
                angulo = rad_to_graus_dec(angulo);
                while(angulo > 180)angulo -= 360;
                while(angulo <= -180)angulo += 360;
                sinal = 1;
                if(angulo<0){ sinal = -1;
                    angulo = -1*angulo;
                }
                graus_sex(angulo);
                if (sinal == -1) gms[0] = -1*gms[0];
                printf("\n %d graus %d minutos %d segundos\n\n", gms[0], gms[1], gms[2]);
                break;
            case 2: printf("\n Entre com o angulo na forma graus minutos segundos\n");
                scanf("%d %d %d", &gms[0], &gms[1], &gms[2]);
                angulo = graus_dec();
                angulo = grad_to_rad(angulo);
                printf("\n %g radianos\n\n", angulo);
                break;
            default: printf("\n opcao invalida : %d\n\n", opt);
                break;
        }
    }
    system("pause");
    return 0;
}

```