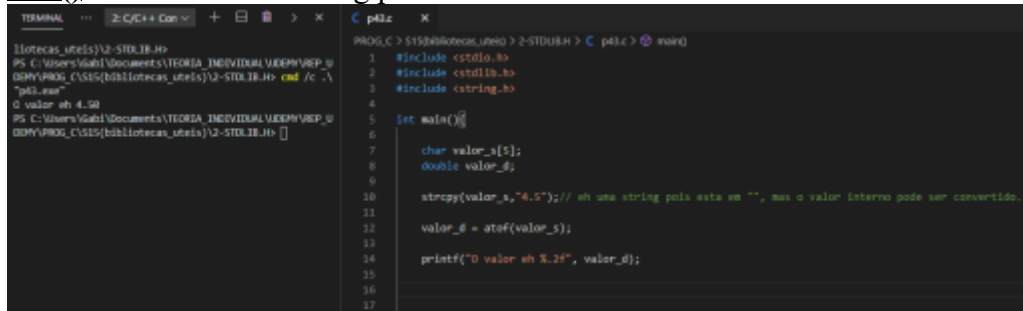


## 2-STDLIB.H

### CONVERSÃO DE STRINGS

**atof()** // converte uma string para double

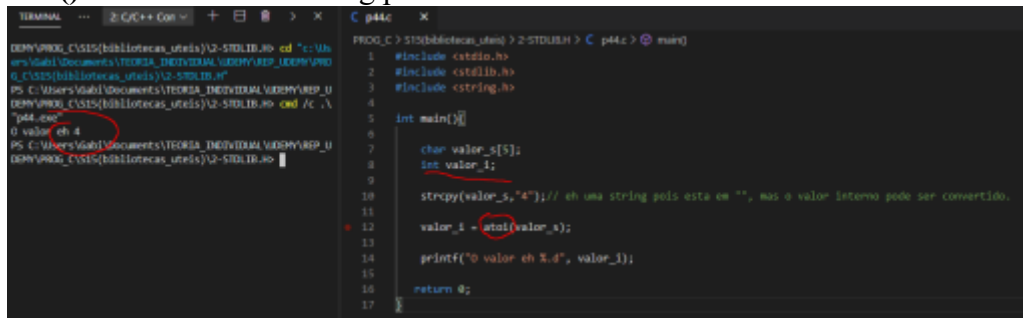


```
TERMINAL ... 2: C/C++ Con ... + - > X C p43.c X
libotecas_uteis\2-STD.LB.H>
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H> cmd /c .\
"p43.exe"
O valor eh 4.50
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H>

PROG_C > 515\libotecas_uteis\2-STD.LB.H > C p43.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(){
6
7     char valor_s[5];
8     double valor_d;
9
10    strcpy(valor_s,"4.5");// eh uma string pois esta em "", mas o valor interno pode ser convertido.
11
12    valor_d = atof(valor_s);
13
14    printf("O valor eh %.2f", valor_d);
15
16
17 }
```

- Estamos pegando a string "4.5" e copiando para a variavel valor\_s.
- Depois pegamos o valor da variavel valor\_s que eh uma string, e estamos utilizando o atof() para converter essa string para o tipo de dado da variavel valor\_d(double).
- So converteu a primeira parte.

**atoi()** // Converte uma string para inteiro.

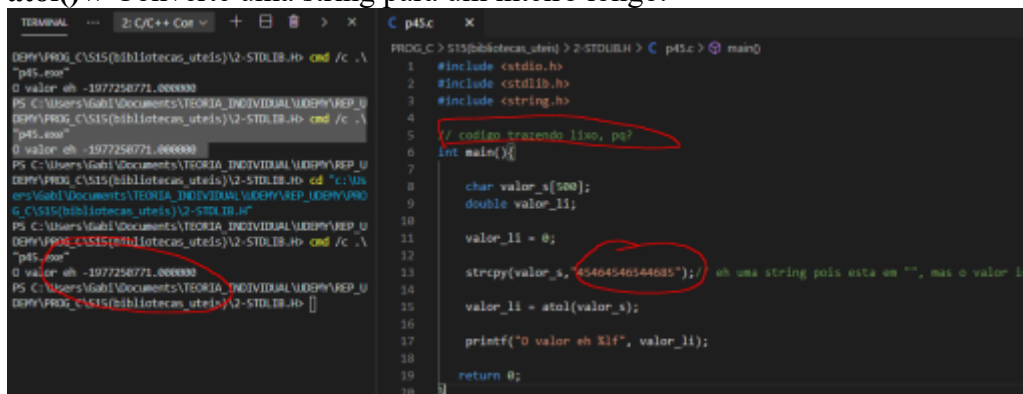


```
TERMINAL ... 2: C/C++ Con ... + - > X C p44.c X
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H> cd "c:\Us
ers\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H> cmd /c .\
"p44.exe"
O valor eh 4
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H>

PROG_C > 515\libotecas_uteis\2-STD.LB.H > C p44.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(){
6
7     char valor_s[5];
8     int valor_i;
9
10    strcpy(valor_s,"4");// eh uma string pois esta em "", mas o valor interno pode ser convertido.
11
12    valor_i = atoi(valor_s);
13
14    printf("O valor eh %d", valor_i);
15
16    return 0;
17 }
```

- Se colocar uma letra, e ele não conseguir converter, eh colocado o valor 0.

**atol()** // Converte uma string para um inteiro longo.



```
TERMINAL ... 2: C/C++ Con ... + - > X C p45.c X
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H> cmd /c .\
"p45.exe"
O valor eh -1977256771.000000
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H> cmd /c .\
"p45.exe"
O valor eh -1977256771.000000
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H> cmd /c .\
"p45.exe"
O valor eh -1977256771.000000
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_U
DEMY\PROG_C\515\libotecas_uteis\2-STD.LB.H>

PROG_C > 515\libotecas_uteis\2-STD.LB.H > C p45.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 // codico trazendo livro, pg?
6 int main(){
7
8     char valor_s[500];
9     double valor_li;
10
11    valor_li = 0;
12
13    strcpy(valor_s,"4564546544685");// eh uma string pois esta em "", mas o valor interno pode ser convertido.
14
15    valor_li = atol(valor_s);
16
17    printf("O valor eh %li", valor_li);
18
19    return 0;
20 }
```

### GERAÇÃO DE SEQUENCIA PSEUDO-ALEATORIA

rand() // gera numero aleatorio

```
TERMINAL ... C/C++ Compile Run ... X
DEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H> cd "c:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H> cd /c .\p46.exe
41
45
51
26
15
47
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H> cd "c:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H> cd /c .\p46.exe
41
45
51
26
15
47
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\515(bibliotecas_uteis)\2-STDLIB.H>

C p46c
PROG_C > 515(bibliotecas_uteis) > 2-STDLIB.H > C p46c > ...
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 //rand()
5 // primeira vez : 41//45//51//26//15//47
6 // segunda vez : 41//45//51//26//15//47
7
8 int main(){
9
10 //Gerar 6 numeros entre 0 e 60
11 for(int i = 0; i < 6; i++){
12     printf("%d\n", rand() % 61); //rand() por padrão inicia em 0.
13 }
14
15 return 0;
16 }
```

- Esta gerando os mesmos numeros aleatorios pois não colocamos o srand().

srand() // inicializa o gerador de numeros aleatorios.

```
PROBLEMS OUTPUT ... C/C++ Compile Run ... X
[p46.c 2020-10-11 12:51:13.631]
p46.c: In function 'main':
p46.c:10:5: error: unknown type name 'time_t'; did you mean 'size_t'?
10 |     time_t t;
    |     ^~~~~~
p46.c:13:22: warning: implicit declaration of function 'time' [-Wimplicit-function-declaration]
13 |     srand((unsigned) time(&t));
    |                      ^~~~~
C p46c
PROG_C > 515(bibliotecas_uteis) > 2-STDLIB.H > C p46c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 //rand()
5 // primeira vez : 41//45//51//26//15//47
6 // segunda vez : 41//45//51//26//15//47
7
8 int main(){
9
10     time_t t;
11
12     //inicializar o gerador
13     srand((unsigned) time(&t));
14
15     //Gerar 6 numeros entre 0 e 60
16     for(int i = 0; i < 6; i++){
17         printf("%d\n", rand() % 61); //rand() por padrão inicia em 0.
18     }
19
20     return 0;
21 }
```

```
stdlib.h ... programa47.c 44
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 //rand()
5
6 //primeira vez 29, 54, 3, 54, 7, 49
7
8 int main(){
9     time_t t;
10
11     //inicializar o gerador
12     srand((unsigned) time(&t));
13
14     //Gerar 6 números entre 0 e 60
15     for(int i = 0; i < 6; i++){
16         printf("%d\n", rand() % 61);
17     }
18     return 0;
19 }
20 }
21

Problems TASKS ... C/C++
<terminated> (exit value: 0) P
53
45
25
49
52
27
```

```
stdlib.h.c | programa47.c 33 Problems Tas
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 //rand()
5
6 //primeira vez 29, 54, 3, 54, 7, 49
7
8 //após inicializar o gerador
9
10 //1 vez 53, 45, 25, 49, 52, 27
11
12 //2 vez
13
14 int main(){
15     time_t t;
16
17     //inicializar o gerador
18     srand((unsigned) time(&t));
19
20
21     //Gerar 6 números entre 0 e 60
22     for(int i = 0; i < 6; i++){
23         printf("%d\n", rand() % 61);
24     }
25     return 0;
26 }
```

## GERENCIAMENTO DE MEMORIA DINAMICA

**malloc()** //Aloca espaço em memoria, mantendo o lixo na memoria.

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

int main(){
    int qtd, *p;

    printf("Informe a quantidade de elementos para o vetor: ");
    scanf("%d", &qtd);

    p = (int*) malloc(qtd * sizeof(int)); //3 x 4 == 12 bytes

    for(int i = 0; i < qtd; i++){
        printf("Informe o valor para a posição %d do vetor: ", i);
        scanf("%d", &p[i]);
    }

    for(int i = 0; i < qtd; i++){
        printf("No vetor 'numeros[%d]' está o valor %d: \n", i, p[i]);
    }

    printf("A variável 'p' ocupa %ld bytes em memória.\n", qtd * sizeof(int));

    //liberar a memória (desalocar)
    free(p);
    p = NULL; //medida de segurança

    return 0;
}
```

```

#include <stdio.h>
#include <stdlib.h>

//Malloc -

int main(){
    int qtd, *p;

    printf("Informe a quantidade de elementos para o vetor: ");
    scanf("%d", &qtd);

    p = (int*)malloc(qtd * sizeof(int)); //3 x 4 == 12 bytes

    if(p){
        printf("A variável 'p' ocupa %ld bytes em memória.\n", qtd * sizeof(int));
    }else{
        printf("Erro: Memória insuficiente!!!");
    }

    //liberar a memória (desalocar)
    free(p);
    p = NULL; //medida de segurança

    return 0;
}

```

**calloc()** //aloca espaço em memoria zerando o espaço

```

int main(){
    int *p;

    //p = (int*)malloc(3 * sizeof(int)); //3 x 4 == 12 bytes //não remove lixo da memória
    p = (int*)calloc(3, sizeof(int)); //3 x 4 == 12 bytes //zera o espaço em memória

    if(p){
        //p[0] = 6;
        //p[1] = 12;
        //p[2] = 34;

        printf("A variável 'p' ocupa %ld bytes em memória.\n", 3 * sizeof(int));
        printf("Valor de p[0] = %d\n", p[0]);
        printf("Valor de p[1] = %d\n", p[1]);
        printf("Valor de p[2] = %d\n", p[2]);
    }else{
        printf("Erro: Memória insuficiente!!!");
    }

    //liberar a memória (desalocar)
    free(p);
    p = NULL; //medida de segurança
}

```

**free()** // Libera memoria, desalocação de memoria

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 //Malloc -
5
6 int main(){
7
8     int qtd, *p;
9
10    printf("Informe a quantidade de elementos para o vetor: ");
11    scanf("%d", &qtd);
12
13    p = (int*)malloc(qtd * sizeof(int)); //3 x 4 == 12 bytes
14
15    if(p){
16        printf("A variável 'p' ocupa %ld bytes em memória.\n", qtd * sizeof(int));
17    }else{
18        printf("Erro: Memória insuficiente!!!");
19    }
20
21    //liberar a memória (desalocar)
22    free(p); I
23    p = NULL; //medida de segurança
24
25
26    return 0;
27 }

```

**realloc()** // modifica o tamanho do espaço alocado em memória.

```

int main(){
    int qtd, *p;

    printf("Informe a quantidade de elementos para o vetor: ");
    scanf("%d", &qtd);

    p = (int*)malloc(qtd * sizeof(int)); //3 x 4 == 12 bytes

    if(p){
        printf("A variável 'p' ocupa %ld bytes em memória.\n", qtd * sizeof(int));

        printf("Informe a quantidade de elementos para o vetor: ");
        scanf("%d", &qtd);

        p = (int*)realloc(p, qtd * sizeof(int));
        I
        if(p){
            printf("A variável 'p' ocupa %ld bytes em memória.\n", qtd * sizeof(int));
        }else{
            printf("Erro: Memória insuficiente!!!");
        }

    }else{
        printf("Erro: Memória insuficiente!!!");
    }
}

```

## PESQUISA E ORDENAÇÃO

**qsort(array, qtd\_elementos, bytes\_cada\_elementos\_memoria, cast\_função\_compara )** //ordena os elementos de um array.

- A função espera que a gente retorne um ponteiro para um VOID.

- O problema de utilizar essa ordenação, eh que temos que fazer uma função de comparação.

```
TERMINAL ... 2: C/C++ Compilador R + [ ] > X
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/powershell

PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY>
cd "c:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H> cmd /c .\"p47.exe"

6
2
9
3
7
1
5
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H> cd "c:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H> cmd /c .\"p47.exe"

1
2
3
5
6
7
9
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H>

C p47.c X
PROG_C > S15(bibliotecas_uteis) > 2-STD LIB.H > C p47.c > [ ] compara(int *, int *)
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int compara(int *x, int *y){
5     if(*x > *y){
6         return 1;
7     }else if(*x == *y){
8         return 0;
9     }else{
10        return -1;
11    }
12 }//Tipo de comparação que o qsort() espera.
13
14
15 int main(){
16
17     int numeros[]={6,2,9,3,7,1,5};
18
19     qsort(numeros, 7, sizeof(int), (void *)compara);
20
21     for(int i = 0; i < 7; i++){
22         printf("%d\n",numeros[i]);
23     }
24
25     return 0;
26 }
```

## ARITIMETICA DE NUMEROS INTEIROS

**abs()** // Valos absoluto.

```
TERMINAL ... 2: C/C++ Compilador R + [ ] > X
PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H> cd "c:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H"
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H> cmd /c .\"p48.exe"

O valor absoluto de 5 eh: 5
O valor absoluto de -10 eh: 10
PS C:\Users\Gabi\Documents\TEORIA_INDIVIDUAL\UDEMY\REP_UDEMY\PROG_C\S15(bibliotecas_uteis)\2-STD LIB.H>

C p48.c X
PROG_C > S15(bibliotecas_uteis) > 2-STD LIB.H > C p48.c > [ ] main()
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main(){
6
7     int valor_a, valor_b;
8
9     valor_a = abs(5);
10    printf("O valor absoluto de 5 eh: %d\n", valor_a);
11
12    valor_b = abs(-10);
13    printf("O valor absoluto de -10 eh: %d\n", valor_b);
14
15
16
17    return 0;
18 }
```

**div()** // Divisão inteira.

```
div_t saida;
/* Returned by 'div'. */
typedef struct
{
    int quot; /* Quotient. */
    int rem; /* Remainder. */
} div_t;
```

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

```
int main(){

    div_t saida;

    saida = div(27, 4);
```

```
printf("O Quociente de 27/4 eh %d\n", saida.quot);
printf("O resto de 27/4 eh %d\n", saida.rem);

saida = div(27, 3);
printf("O Quociente de 27/3 eh %d\n", saida.quot);
printf("O resto de 27/3 eh %d\n", saida.rem);

return 0;
}
```

=saida

O Quociente de 27/4 eh 6  
O resto de 27/4 eh 3  
O Quociente de 27/3 eh 9  
O resto de 27/3 eh 0