



Em C, temos a função de classificação geral **qsort** que implementa o algoritmo de ordenação rápida. A descrição da biblioteca C descreve-o como

```
void qsort(void *base, size_t nitems, size_t size, int (*compare)
(const void *, const void*))
```

É uma função que recebe como argumentos

- Um apontador para o array de elementos (do tipo indeterminado): **base**
- Um int do número de itens no array: **nitems**
- O tamanho de um elemento no array: **size**
- Um apontador para uma função que compara dois elementos (especificados por dois apontadores gerais) e retorna um int: **compare**

Por sua vez, esta função de comparação deve obedecer a este requiste:

- recebe dois ponteiros para elementos
- retorna um int especificando qual elemento é 'maior'.

Como exemplo, onde a array a ser classificada é um array de int:

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

// Comparison function
int compare(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}

int main() {
    int arr[] = {10, 5, 4, 6, 9};
    int n = sizeof(arr) / sizeof(arr[0]);

    qsort(arr, n, sizeof(int), compare);

    printf("Following is the sorted array: ");
```

```

int i;
for (i = 0; i < n; ++i) {
    printf("%d ", arr[i]);
}
printf("\n");
return 0;
}

```

Que tem como output:

Following is the sorted array: 4 5 6 9 10

Vamos fazer o mesmo, mas em vez de **qsort** vamos usar **bsort**, ou seja, bubble-sort. O algoritmo para bubble-sort é o seguinte:

bsort algorithm:

- Set the flag **change** to **false**,
- In a for-loop from **i=0** to **n-2**, check if **element[i] > element[i+1]**, if so exchange them and raise the flag **change** to **true** that indicates a change has taken place in this iteration,
- Repeat until **change** is **false**.

Escreva um programa MIPS Assembly que implemente a função **bsort** e teste-a com uma matriz de int e uma matriz de float. Arrays a serem usadas:

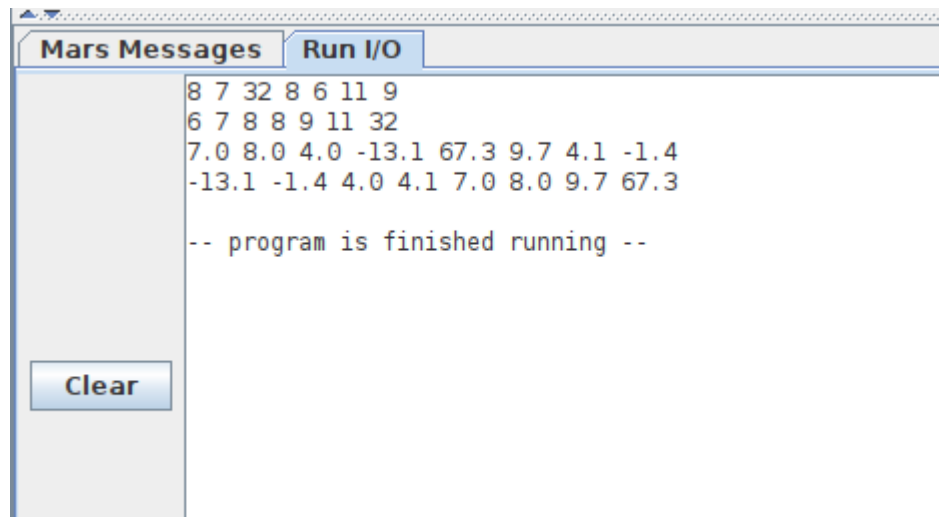
```

.data
myintsarray: .word 8 7 32 8 6 11 9
myfloatsarray: .float 7.0 8.0 4.0 -13.1 67.3 9.7 4.1 -1.4

```

**bsort**, tal como **qsort**, recebe estes argumentos:

- Um **apontador** para o array de elementos (do tipo indeterminado)
- Um **int** com o número de elementos
- Um **int** especificando o tamanho (bytes) de um elemento no array
- Um **apontador** para a função que compara dois elementos



- Provavelmente é uma boa ideia de começar escrever um programa em C.
- As funções e a main têm de usar no mínimo um registo \$t e \$s
- Grupos de uma (1) ou duas (2) pessoas.
- Entregue um (1) ficheiro (**aXXXXXaYYYYY.asm**) na Tutoria Electronica. (Não é necessário para os dois membros do grupo entregar um ficheiro).
- Deadline: **18 maio 2025, 21h00** (Hora de Lisboa).
- Escreve os dois (2) nomes como comentário na primeira linha do ficheiro.