

Programação 2023/2024 – 2º Semestre

Aula de Problemas 4

Tente resolver todos os exercícios antes da aula de problemas. Bom trabalho!

1. Indique o que é escrito no ecrã quando se executa o seguinte programa:

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.
4.  void x(char * a, char *b)
5.  {
6.      char *c;
7.
8.      c = a;
9.      a = b;
10.     b = c;
11.     *c = *c + (*a-*b)/2;
12.     *b = '\0';
13. }
14.
15. void y(char ** a, char ** b)
16. {
17.     char *c;
18.
19.     c = *b;
20.     *b = *a;
21.     *a = c+1;
22. }
23.
24. int main()
25. {
26.     char s1[] = "1234567890";
27.     char s2[] = "abcdefghij";
28.     char * a, *b;
29.
30.     a = s2;
31.     b = s2+5;
32.
33.     x(&s1[7], s1);
34.     y(&a, &b);
35.     *a = ' ';
36.
37.     printf("%s\n", s1);
38.     printf("%s\n", s2);
39.     exit(0);
40. }
```

2. Desenvolva uma função (**my_strrchr**) que recebe como argumento uma string (**str**) e um caractere (**c**) e retorna um ponteiro para a última ocorrência do caractere **c** na string **str**. Esta função é equivalente à função **strrchr** da biblioteca do C. Não use a função **strrchr**. Se o caractere não for encontrado deverá ser retornado NULL. O cabeçalho da função é:

```
char * my_strrchr(char *str, char c)
```

3. Escreva um programa que pesquisa uma única palavra num ficheiro de texto, apresentando todas as linhas do ficheiro que contenham a palavra precedidas do número de linha do ficheiro e do caractere ':', tal como no comando Linux "grep -n". O programa deve receber a palavra a pesquisar e o nome do ficheiro, respectivamente, como 1º e 2º argumentos da linha de comando.

- a. Os dois parâmetros são introduzidos como argumentos de execução do programa. Devem recorrer à seguinte definição da função main:

```
int main(int argc, char *argv[])
```

- b. argc é o número de argumentos (o nome do programa também conta como argumento), argv[0] aponta para o nome do programa e as restantes posições para cada um dos argumentos.
- c. Utilize a função **strstr** para realizar a procura.
4. Pretende-se desenvolver um programa que lê contactos telefónicos de um ficheiro chamado 'phonebook.txt', armazena-os num vector de estruturas, e permite a procura de contactos pelo nome. Utilize alocação de memória estática para resolver este exercício. Pode admitir que a lista telefónica não terá mais de 1000 entradas. Pode admitir que o nome não terá mais de 100 caracteres. O utilizador deverá poder escrever um nome e será apresentado no ecrã todos os contactos que contenham o nome escrito. Se pesquisar o nome "*", deve mostrar toda a lista telefónica. Para tal:

- a. Defina o tipo de dados phoneEntry que representa um contacto. Cada um dos contactos deverá ser armazenado numa estrutura contendo:

- Nome;
- Número de telefone.

- b. Desenvolva a função void load_phonebook(FILE *fp, phoneEntry *phonebook) que lê um ficheiro contendo contactos telefónicos e armazena-os num vector. Cada linha do ficheiro contém um nome e um número de telefone separados por uma vírgula. Por exemplo:

```
John,910000000
```

- c. Desenvolva a função void print_number(phoneEntry *phonebook, char *pname) que imprime todos os contactos que contenham o nome pname escrito.