Universidade Federal do Maranhão
Departamento de Informática
Ciência da Computação
Linguagem de programação
Prof.: Francisco Glaubos


# Lista 2

strings, ponteiros, alocação dinâmica, structs e arquivos

**Questão 1:**

Escreva um programa C para converter uma string com letras minúsculas em uma string com letras maiúsculas.

**Questão 2:**

Escreva um programa em C para verificar se uma string é palíndromo ou não.

**Questão 3:**

Escreva um programa em C para inverter a ordem das palavras em uma determinada string.

**Questão 4:**

Escreva um programa em C para encontrar a primeira ocorrência de uma palavra em uma determinada string.

**Questão 5:**

Escreva um programa em C para remover todos os espaços em branco extras de string de entrada.

**Questão 6:**

Escreva um programa em C para remover todos os caracteres repetidos de uma determinada string.

**Questão 7:**

Escreva um programa em C para contar as ocorrências de uma palavra em uma determinada string.

**Questão 8:**

Crie uma função em C para trocar os valores de dois números passados por referência.

Ex.:
Antes da chamada da função: a=1 e b=2
Após a chamada da função: b=1 e a=2

**Questão 9:**

Escreva um programa C para buscar um elemento em um vetor, usando ponteiros.

**Questão 10:**

Escreva um programa em C para imprimir todos as letras do alfabeto usando um ponteiro.

**Questão 11:**

Qual será a saída do seguinte programa em C?

```c
#include<stdio.h>
int main()
{
  char *ptr = "helloworld";
  printf(ptr + 4);
  return 0;
}
```

a) oworld b) world c) hell d) hello

**Questão 12:**

Qual será a saída do seguinte programa em C?

```c
#include<stdio.h>
int main()
{
  char *ptr = "auladelp";
  printf("%c\n",*&*ptr);
  return 0;
}
```

a) Endereço de 'a' b) Erro de compilação c) a d) Erro de execução

**Questão 13:**

Qual será a saída do seguinte programa em C?

```c
#include<stdio.h>
int main(){
  int a = 25, b;
  int *ptr, *ptr1;
  ptr = &a;
  ptr1 = &b;
  b = 36;
  printf("%d %d",*ptr, *ptr1);
  return 0;
}
```

a) 25 45632845

b) Erro de execução

c) Erro de compilação

d) 25 36

**Questão 14:**

Qual será a saída do seguinte programa em C?

```c
#include<stdio.h>
int main(){
  int a = 36;
  int *ptr;
  ptr = &a;
  printf("%u %u", *&ptr , &*ptr);
  return 0;
}
```

a) Endereço Valor

b) Valor Endereço

c) Endereço Endereço

d) Erro de compilação

**Questão 15:**

Qual será a saída do seguinte programa em C?

```c
#include<stdio.h>
int main(){
  int i = 25;
  int *j;
  int **k;
  j = &i;
  k = &j;
  printf("%u %u %u ",k,*k,**k);
  return 0;
}
```

a) endereço endereço valor

b) endereço valor valor

c) endereço endereço endereço

d) erro de compilação

**Questão 16:**

Crie um programa em C para simular um carrinho de compras com tamanho flexível.

Utilizando alocação dinâmica de memória, crie funções para adicionar e remover nomes de produtos em uma variável char* carrinho. A capacidade do carrinho deve ser inicialmente igual a 0, e deve ser aumentada em 1 unidade antes de cada nova adição de produto.

**Questão 17:**

Escreva um programa em C para manter registros e realizar análises estatísticas para uma turma de 20 alunos. As informações de cada aluno contêm ID, nome, sexo, pontuação dos testes (2 testes por semestre) e pontuação total
O programa solicitará que o usuário escolha a operação de registros em um menu, como mostrado abaixo:

```
==========================================
                Menu
==========================================
1. Adicionar registros de estudante

2. Deletar registros de estudante

3. Atualizar registros de estudante

4. Ver registros de todos os estudantes

5. Mostrar aluno com maior nota total

6. Mostrar aluno com menor nota total

7. Encontrar aluno por ID

8. Ordenar os registros por pontuacao total
```

Entre com a sua opção:

Obs.: Todos os registros de aluno devem ser armazenados em vetores de *structs*

**Questão 18:**

Crie uma struct para representar uma fração, na qual um campo representa o numerador e outro o denominador da fração.
Em seguida crie uma função para somar duas frações e exibir a fração do resultado. Seu programa solicitará que o usuário insira a fração 1 e a fração 2. O numerador e o denominador de

cada fração são inseridos separadamente pelo espaço. Veja o exemplo de saída abaixo:

Digite a fração 1 (denominador do numerador): 1 2
Digite a fração 2 (denominador do numerador): 2 5
Resultado: 9/10

**Questão 19:**

Defina um tipo de estrutura para representar um ponto através de suas coordenadas cartesianas. Em seguida, crie uma função para calcular e retornar a distância euclidiana entre dois pontos fornecidos como entrada.

Dica: a distância entre dois pontos P e Q, é dada pela seguinte fórmula:

$d_{qp} = \sqrt{(x_q - x_p)^2 + (y_q - y_p)^2}$

**Questão 20:**

Crie uma struct para representar uma Pessoa, com nome (até 100 caracteres), endereço (até 200 caracteres) e telefone; em seguida, crie uma agenda para armazenar o contato de até $n$ pessoas (o espaço da agenda deverá ser alocado dinamicamente).

Em seguida, crie uma função `void addPessoa(Pessoa* agenda, int n)` que adicione uma pessoa à agenda;

**Questão 21:**

A partir do exercício anterior, crie uma função para buscar uma pessoa na agenda (pelo nome) e retornar o número do seu telefone.

**Questão 22:**

Qual será a saída do seguinte programa em C?

```c
int main() {
  int v[]={0};
  v[0]=1;
  free(v);
  printf("%d", v[0]);
return 0;
}
```

a) erro de compilação

b) erro de execução

c) 0

d) 1

**Questão 23:**

Quando fopen() não é capaz de abrir um arquivo, ele retorna:

a) EOF

b) NULL

c) Ocorre erro de execução

d) Ocorre erro de compilação

**Questão 24:**

Escreva um programa em C para contar caracteres, palavras e linhas de um arquivo de texto.

**Questão 25:**

Escreva um programa em C para remover uma palavra de um arquivo de texto.

**Questão 26:**

Escreva um programa C para imprimir na tela o seu próprio código-fonte.

**Questão 27:**

Escreva um programa C para remover linhas vazias de um arquivo de texto.

**Questão 28:**

Escreva um programa em C para localizar e substituir uma palavra em um arquivo de texto.

**Questão 29:**

Escreva um programa C para renomear um arquivo usando a função *rename()*.

**Questão 30:**

Qual será a saída do seguinte programa em C?

```c
#include<stdio.h>
int main()
{
  int EOF = 0;
  printf("%d", EOF);
  return 0;
}
```

a) -1

b) 0

c) 1

d) Erro de compilação

**Questão 31:**

Qual será a saída do seguinte programa em C? Paral tal, considere que o arquivo data.txt contém o seguinte conteúdo: *Aula de LP.*

```c
#include<stdio.h>
int main()
{
  unsigned char ch;
  FILE *fp;
  fp = fopen("data.txt", "r");
  while ((ch = fgetc(fp)) != EOF)
  {
    printf("%c", ch);
  }
  printf(" Obrigado.");
  fclose(fp);
  return 0;
}
```

a) erro de compilação

b) nada será mostrado

c) "Aula de LP. Obrigado."

d) "Aula de LP". E o loop continua infinitamente.

**Questão 32:**

Qual será a saída do seguinte programa em C? Paral tal, considere que o arquivo data.txt contém o seguinte conteúdo: *Aula de LP.*

```c
#include<stdio.h>
int main()
{
  char ch;
  FILE *fp;
  fp = fopen("data.txt", "w");
  while ((ch = fgetc(fp)) != EOF)
  {
    printf("%c", ch);
  }
  printf(" Obrigado.");
  fclose(fp);
  return 0;
}
```

a) erro de compilação

b) "Obrigado"

c) "Aula de LP. Obrigado."

d) "Aula de LP". E o loop continua infinitamente.

**Questão 33:**

Qual o tipo de retorno da função *malloc()*?

a) void*

b) ponteiro do tipo da memória alocada

c) void**

d) int*

**Questão 34:**

Qual o problema do seguinte código?

```c
#include<stdio.h>
int main()
{
    int *p = (int *)malloc(sizeof(int));

    p = NULL;

    free(p);
}
```

a) Erro de compilação: free() não pode ser aplicado a um ponteiro que aponta para NULL

b) Vazamento de memória

c) O programa pode falhar quando free() é chamado para ponteiro NULL.

d) Ponteiro oscilante: não apontando para uma espaço de memória não existente