Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Escola Politécnica – Curso de Bacharelado em Engenharia de *Software*
Disciplina: Programação de *Software* Básico – Professor: Roland Teodorowitsch
14 de agosto de 2025

Lista de Exercícios 2 (2025/2)

Este trabalho consiste em resolver a lista de exercícios das páginas a seguir, em C.
Para a entrega devem ser seguidas as seguintes regras:

- criar um arquivo compactado no formato ZIP com o nome do aluno no formato *camelHump* (por exemplo, para João Pedro da Silva, usar `JoaoPedroDaSilva.zip`), **SEM SUBDIRETÓRIOS** e **APENAS COM OS ARQUIVOS DE CÓDIGO-FONTE** (NÃO envie quaisquer outros arquivos, como, por exemplo, arquivos .class, .ctxt, README.txt, .o ou executáveis);
- o código-fonte deve ser **ADEQUADAMENTE INDENTADO**;
- o arquivo compactado deve conter programas em C para resolver cada um dos exercícios, salvando o código-fonte em um arquivo com o nome `Exercicio` seguido do número do exercício com **TRÊS dígitos** (por exemplo, `Exercicio001.c`, `Exercicio002.c`, ..., `Exercicio050.c`, ..., `Exercicio101.c`, ...);
- **ATENÇÃO:** os exercícios NÃO seguem necessariamente uma sequência contínua, então tome cuidado de **USAR O NÚMERO CORRETO DO EXERCÍCIO NO RESPECTIVO ARQUIVO DE CÓDIGO-FONTE**;
- **NÃO USAR ACENTOS NO NOME DE ARQUIVOS E DE FUNÇÕES** ;
- no início de cada arquivo em C, incluir um comentário informando o nome do arquivo, o nome do autor, a finalidade do programa e a versão (ou data) de criação (ou atualização);
- quando houver dados a serem lidos, **LER OS DADOS SEMPRE NA MESMA ORDEM EM QUE ELES SÃO CITADOS NO ENUNCIADO**, escolhendo os tipos numéricos adequadamente (se houver dúvida entre usar um tipo inteiro ou ou tipo real, use os exemplos de entradas e saídas que aparecem após cada exercício);
- **ESCREVER OS RESULTADOS SEMPRE NA MESMA ORDEM EM QUE ELES SÃO CITADOS NO ENUNCIADO**, escolhendo os tipos numéricos adequadamente (**NÚMEROS REAIS DEVEM SER IMPRESSOS SEMPRE COM 4 CASAS DECIMAIS**, a não ser que seja explicitamente indicado de outra forma);
- na versão final, tomar o cuidado de **NÃO IMPRIMIR NADA DIFERENTE DA SAÍDA ESPERADA** (não devem aparecer, por exemplo, mensagens pedindo que o usuário forneça ou digite determinado valor no terminal);
- a entrega deverá ser feita no dia e horário informado pelo professor em sala de aula e/ou definida na opção de entrega da plataforma moodle da PUCRS.

1. Escreva uma função em C para um caixa de banco, que recebe um valor inteiro `r` (maior ou igual a zero) e determina o número de notas de 100, 50, 10, 5 e 1 reais necessário para pagar a quantia `r`. Faça de forma que o número de notas usado seja o menor possível, retornando as quantidades de notas **por referência**.

   Esta função deve ser inserida, e deve funcionar corretamente, no seguinte programa:

```c
#include <stdio.h>

void papelMoeda(unsigned r, unsigned *n100, unsigned *n50, unsigned *n10, unsigned *n5, unsigned *n1);

int main() {
  unsigned r, n100, n50, n10, n5, n1;
  scanf("%u", &r);
  papelMoeda(r, &n100, &n50, &n10, &n5, &n1);
  printf("%u␣%u␣%u␣%u␣%u\n", n100, n50, n10, n5, n1);
  return 0;
}
```

   Observe que o código acima apresenta o protótipo da função.

   Exemplo(s):

Teste	Entrada	Saída
1	166	1 1 1 1 1
2	165	1 1 1 1 0
3	161	1 1 1 0 1
4	156	1 1 0 1 1
5	116	1 0 1 1 1
6	66	0 1 1 1 1
7	54321	543 0 2 0 1
8	12345	123 0 4 1 0
9	12347	123 0 4 1 2
10	999	9 1 4 1 4

2. Escreva uma função em C que receba um tempo em total de segundos desde a meia-noite (um valor maior ou igual a 0 e menor ou igual a 86399), retornando **por referência** o número de horas, de minutos e de segundos correspondentes.

   Esta função deve ser inserida, e deve funcionar corretamente, no seguinte programa:

```c
#include <stdio.h>

void hms(unsigned seg, unsigned *h, unsigned *m, unsigned *s);

int main() {
  unsigned seg, h, m, s;
  scanf("%u", &seg);
  hms(seg, &h, &m, &s);
  printf("%u␣%u␣%u\n", h, m, s);
  return 0;
}
```

   Observe que o código acima apresenta o protótipo da função.

   Exemplo(s):

Teste	Entrada	Saída
1	0	0 0 0
2	3661	1 1 1
3	7384	2 3 4
4	18243	5 4 3
5	73840	20 30 40
6	80421	22 20 21
7	54977	15 16 17
8	36001	10 0 1
9	15000	4 10 0
10	86399	23 59 59

3. Escreva uma função em C que receba um vetor de inteiros, mais a quantidade de valores presentes nele, e retorne um número inteiro correspondente à quantidade de valores que aparecem mais de uma vez. Por exemplo, se o vetor for [1, 2, 3, 4, 5, 3, 6, 7, 2, 8], o retorno será 2 (2 números se repetem, 2 e 3).

   Esta função deve ser inserida, e deve funcionar corretamente, no seguinte programa:

```c
#include <stdio.h>

#define MAX 100

int conta_repetidos(int *vet, int tam);

int main() {
  int vet[MAX], tam;
  scanf("%d", &tam);
  if ( tam > MAX )
     return 1;
  for (int i=0; i < tam; ++i)
      scanf("%d", &vet[i]);
  printf("%d\n", conta_repetidos(vet, tam));
  return 0;
}
```

Observe que o código acima apresenta o protótipo da função. Neste código o tamanho máximo do vetor é limitado. Basicamente, lê-se do terminal o número de elementos do vetor e cada um desses elementos, chamando a função e imprimindo o resultado de sua execução.

Exemplo(s):

Teste	Entrada	Saída
1	10 1 2 3 4 5 3 6 7 2 8	2
2	12 1 2 3 4 5 6 7 8 9 10 11 12	0
3	8 1 1 1 1 1 0 1 1	1
4	15 1 2 3 4 5 1 2 3 6 7 1 2 8 1 9	3
5	14 1 2 3 4 5 3 4 2 6 4 3 2 2 1	4
6	15 1 2 3 4 5 6 1 1 2 7 8 4 5 9 9	5
7	10 1 1 1 1 1 1 1 1 1 1	1
8	9 5 4 3 2 1 2 3 4 5	4
9	10 1 1 2 2 3 3 4 4 5 5	5
10	10 1 2 3 3 3 2 2 1 1 1	3