

7 DE JULHO DE 2014

**Professor:** Rosane Minghim  
**Estagiário PAE:** Carlos Elias Arminio Zampieri

## Trabalho 6

### Conceitos Gerais do Conteúdo Estudado

Data da Entrega: 11/07/2014 - até 23h59min

**ATENÇÃO:** Leia as informações a seguir antes de iniciar o trabalho:

- Os algoritmos abaixo devem utilizar comandos de Conceitos Gerais do Conteúdo Estudado. **Para todos os exercícios devem ser elaborados os códigos e os programas que os executam em linguagem C.** Lembre de nomear seus arquivos indicando qual o número da questão.
- Os padrões de entrada e saída esperados pelo SSP são informados após cada questão e precisam ser seguidos exatamente como nos exemplos (formatação de saída, textos, mensagens, etc) para uma avaliação positiva do trabalho. Lembre que não é necessário a impressão de mensagens pedindo ao usuário que informe os dados, apenas os métodos de leitura (scanf).
- Os códigos dos programas (extensão “.c”) deverão ser salvos em arquivos separados por exercício e submetidos ao sistema SSP observando qual o trabalho e exercício relacionado. O site, conforme anteriormente apresentado, é: <http://ssp.icmc.usp.br>.

1. Implementar um subprograma em C para encontrar o conjunto dos primeiros **K** números primos. Use um vetor de inteiros para armazenar o resultado de retorno. Faça um programa principal para testar sua função onde o usuário possa informar a quantidade de números primos desejada e que imprima em tela o vetor com os resultados. O programa deve parar quando a quantidade de número digitada for igual a -1.

Exemplo de entrada e saída no sistema de submissão:

Entrada:

```
3      -{K=3}
5      -{K=5}
-1     -{K=-1 - encerra o programa}
```

Saída:

```
P1: 2 P2: 3 P3: 5      -{Saída para K=3}
P1: 2 P2: 3 P3: 5 P4: 7 P5: 11    -{Saída para K=5}
```

\* Todo texto entre -{} é um comentário explicativo e não deve aparecer nos testes.

2. Desenvolva um programa em C que receba via teclado dois valores, o primeiro servindo de indicador de operação e o segundo correspondendo ao raio de uma circunferência. Caso o primeiro valor seja igual a 1, chamar um subprograma que receba como parâmetro o raio e retorne a área desta circunferência. Se o valor lido for 2, chamar um subprograma que receba por parâmetro o raio e retorne o perímetro da circunferência, e caso o valor seja 3, ler do teclado um segundo valor de raio para outra circunferência e chamar um subprograma informando como parâmetros os dois raios e faça a comparação das duas figuras geométricas. Caso a área e perímetro sejam

iguais retorne valor **1**, caso contrário **0**. Faça um programa principal que permita ao usuário realizar um número indefinido de operações encerrando quando o valor da operação foi igual a **-1**. Caso o valor da operação seja diferente de **-1** e não esteja listado nas operações, imprimir em tela a mensagem **“Operacao Invalida”**. Para a opção 3 imprima em tela no programa principal se são iguais ou não conforme exemplo.

Exemplo de entrada e saída no sistema de submissão:

Entrada:

```
2    -{Operação perímetro}
4.32 -{Entrada valor raio}
1    -{Operação área}
4.32 -{Entrada valor raio}
3    -{Operação comparação}
8.52 -{Entrada valor raio 1}
8.52 -{Entrada valor raio 2}
3    -{Operação comparação}
8.52 -{Entrada valor raio 1}
4.0  -{Entrada valor raio 2}
8    -{Operação Inválida}
-1   -{Condição de parada}
```

Saída:

```
Perimetro: 27.14    -{Resultado operação 2}
Area: 58.63        -{Resultado operação 1}
As figuras de raio 8.52 e 8.52 sao iguais    -{Resultado operação 3}
As figuras de raio 8.52 e 4.00 sao diferentes -{Resultado operação 3}
Operacao Invalida -{Resultado operação 8}
```

\* Todo texto entre `-{ }` é um comentário explicativo e não deve aparecer nos testes.

3. Implementar um subprograma que receba como parâmetros os dois lados de um quadrilátero (L1 e L2) com as mesmas unidades de medida e retorne a **área**, o **perímetro** e o **tipo de figura geométrica** (**0** quadrado e **1** retângulo). Escreva um programa principal que leia do teclado os lados do quadrilátero até o usuário informar valor **-1** para o primeiro lado. Use passagem de parâmetros por referência.

Exemplo de entrada e saída no sistema de submissão:

Entrada:

```
10.10 -{Lado 1 figura 1}
5.65  -{Lado 2 figura 1}
4.22  -{Lado 1 figura 2}
4.22  -{Lado 2 figura 2}
-1    -{Condição de parada}
```

Saída:

```
Area: 57.07 Perimetro: 31.50    -{Resultado figura 1}
Eh um retangulo    -{Resultado figura 1}
Area: 17.81 Perimetro: 16.88    -{Resultado figura 2}
Eh um quadrado    -{Resultado figura 2}
```

\* Todo texto entre `-{ }` é um comentário explicativo e não deve aparecer nos testes.

4. Escreva um subprograma em C que receba como parâmetro uma cadeia de caracteres e verifique se ela é um palíndromo e retorne **1** em caso positivo e **0** em caso negativo. Palíndromo é uma

palavra ou sentença que pode ser lida da mesma forma da esquerda para a direita ou da direita para esquerda (EX: radar é um palíndromo, radares não). Implemente um programa principal que permita fazer verificações até o usuário informar uma cadeia de caracteres contendo o texto “FIM”.

Exemplo de entrada e saída no sistema de submissão:

Entrada:

```
radar      -{Palíndromo}
reviver    -{Palíndromo}
macaco     -{Não palíndromo}
sopapos    -{Palíndromo}
pedra      -{Não palíndromo}
FIM        -{Condição de parada}
```

Saída:

```
A cadeia radar eh palindromo      -{Resultado 1ª cadeia}
A cadeia reviver eh palindromo     -{Resultado 2ª cadeia}
A cadeia macaco nao eh palindromo  -{Resultado 3ª cadeia}
A cadeia sopapos eh palindromo     -{Resultado 4ª cadeia}
A cadeia pedra nao eh palindromo   -{Resultado 5ª cadeia}
```

\* Todo texto entre -{} é um comentário explicativo e não deve aparecer nos testes.