

Métodos e Técnicas de Programação

:: Avaliação continuada P6 ::

Prof. Igor Peretta

ENTREGA: até 14/nov/2018

Contents

1	Programas a serem entregues	1
1.1	P6.c	1
1.1.1	Dicas	1
1.1.2	Testes	3
2	Informações importantes	3

1 Programas a serem entregues

Os programas a serem entregues precisam seguir o nome da seção em que são descritos, não sendo aceitos programas com outros nomes.

1.1 P6.c

A partir de uma `struct "ponto"` (coordenadas `x` e `y`), escreva um programa que peça ao usuário um número N de pontos desejado e que mostre as N coordenadas dos pontos que dividem uma circunferência de raio um em partes iguais.

1.1.1 Dicas

1. Você pode usar como referência o código exemplo que foi disponibilizado (`mem_din.c`).
2. Não se esqueça do `getchar()` para limpar o ENTER do buffer de teclado após o `scanf()` na captura das opções do usuário.

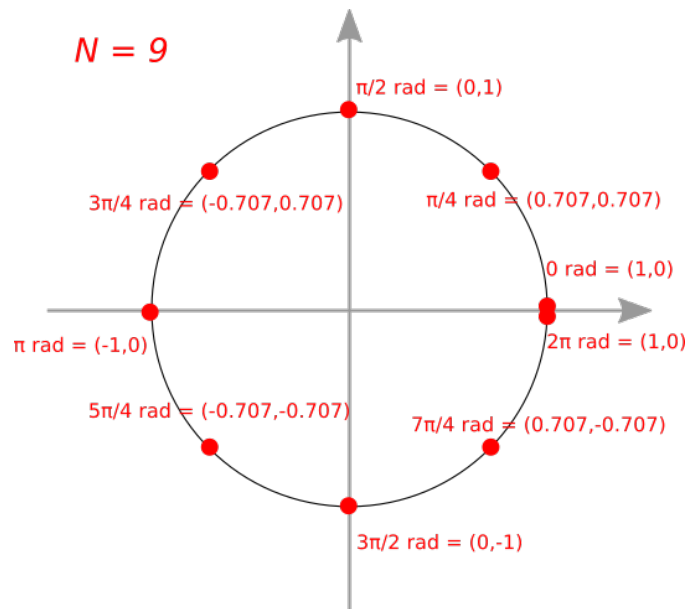


Figure 1: Exemplo gráfico de saída para 9 pontos.

3. As bibliotecas a serem usadas: `stdio.h`, `stdlib.h` e `cmath.h`.
4. Não esqueça do `free()` ao final do programa para evitar vazamento de memória.
5. Ao compilar seu código, se estiver tendo problemas com os comandos da biblioteca `math.h`, basta incluir a flag `-lm` no comando de compilação. Exemplo: `gcc -o P7 P7.c -lm`.
6. Para gerar os pontos referentes ao círculo unitário, gere uma divisão do círculo em ângulos (de 0 a 2π), usando `sin` e `cos` para transpor para o plano xy :

```
x = cos(i*2.0*M_PI/(N-1));
y = sin(i*2.0*M_PI/(N-1));
```

7. Para mostrar os pontos, use o especificador `%.3lf` para deixar os números com apenas três casas decimais.

1.1.2 Testes

Os seguintes testes devem ser efetuados em sequência a partir de uma nova execução do programa:

- "2" retorna "(1.000, 0.000) (1.000, -0.000)"
- "5" retorna "(1.000, 0.000) (0.000, 1.000) (-1.000, 0.000) (-0.000, -1.000) (1.000, -0.000)"
- "7" retorna "(1.000, 0.000) (0.500, 0.866) (-0.500, 0.866) (-1.000, 0.000) (-0.500, -0.866) (0.500, -0.866) (1.000, -0.000)"
- "9" retorna "(1.000, 0.000) (0.707, 0.707) (0.000, 1.000) (-0.707, 0.707) (-1.000, 0.000) (-0.707, -0.707) (-0.000, -1.000) (0.707, -0.707) (1.000, -0.000)"

2 Informações importantes

É necessário criar em sua conta do github um repositório com o nome 'MTP-2018-2'. É nesse repositório que você dar *upload* do(s) seu(s) código-fonte(s) (ex. arquivos `P1.c`, `P2.c` etc.), não sendo desejado nenhum executável ou arquivo de apoio de projetos.

Em todo programa que você fizer, comece com seu nome e matrícula como comentários. Se não constar essas informações nos arquivos enviados para seu repositório no Github, os programas serão **desconsiderados**.

Mantenha seu código limpo. Não use comandos como `system(pause)` ou `#include<conio.h>` pois são específicos do sistema operacional Windows. Se usá-los, seu código-fonte poderá não compilar, invalidando sua entrega.