

# Décimas Olimpíadas Nacionais de Informática

## 1º Problema

### Quadrados e círculos

Imagine um quadrado e um círculo desenhados num plano. Pode ser que nem se toquem, pode ser que um esteja dentro do outro, pode ser que se sobreponham apenas parcialmente. Pois bem, o que se pretende é um programa para calcular a área comum às duas figuras. O quadrado é especificado pelas coordenadas cartesianas de dois vértices opostos, e o círculo pelas coordenadas do centro e pelo raio (tudo números reais).

O programa deve ter um funcionamento iterativo e interactivo. No início, o programa afixa no ecrã uma mensagem de apresentação. Depois, em cada passo do ciclo, pede e aceita os valores numéricos que definem o quadrado, depois os que definem o círculo, e a seguir mostra no ecrã o resultado, com duas casas decimais. A seguir, pergunta ao utilizador se quer fazer mais cálculos, e se a resposta for “S”, “s”, “Y” ou “y”, ou ainda a palavra “sim” ou “yes” em qualquer combinação de maiúsculas ou minúsculas, então volta a pedir os dados e a apresentar os resultados, etc. Caso contrário, termina com uma mensagem de despedida.

Nota: o funcionamento do programa deve ser exactamente como descrito, ficando a escolha do texto das mensagens ao seu cuidado. O júri apreciará a clareza e a simplicidade do funcionamento.

## 2º Problema

### Números primos duplos

Um número primo é um número natural maior que 1 que só é divisível por ele próprio e por 1. A sequência dos números primos é bem conhecida: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ... Curiosamente, alguns destes números primos são formados pela “justaposição” de dois outros números primos: por exemplo 23 e 37. Estes são pequenos, e vê-se logo. Mas há com certeza outros maiores, como por exemplo 1277, 1913, 27737, 29881, mais difíceis de apanhar.

Pois bem, chama-se número primo duplo um número primo que se pode obter por justaposição de dois outros números primos, e o seu programa vai servir para calcular números primos duplos.

Para isso, o programa vai ler um ficheiro de texto INPUT.TXT, que contém uma única linha, na qual existe um único número inteiro maior que zero,  $i$ , e escrever um ficheiro de texto OUTPUT.TXT também com uma única linha com um único número. Este número é precisamente o  $i$ -ésimo número primo duplo.

Nota: o seu programa não deve usar nem o terminal, nem outros ficheiros, para além dos dois mencionados.

### Requisitos técnicos:

1. Os programas devem ser escritos em Pascal ou C/C++, e correr numa janela de DOS.
2. Os programas devem realizar a tarefa pedida, e nada mais, e não devem usar outros recursos para além dos estritamente necessários para isso.
3. Os programas-fonte devem ter o nome QC, para o primeiro problema e PD, para o segundo, com a extensão da linguagem usada; os executáveis chamam-se qc.exe e pd.exe.
4. Os programas Pascal devem estar num único ficheiro; os programas C/C++ podem ser organizados em vários módulos, e o requisito do número anterior aplica-se ao ficheiro que contém a função main.
5. Cada ficheiro-fonte deve começar com um comentário com o nome do autor.
6. Deve ser fornecido para cada problema, um ficheiro de texto, qc.txt e pd.txt, onde o autor descreverá brevemente a estratégia usada e as principais dificuldades (se tiver havido dificuldades).

## Recomendações

1. Estes problemas devem ser resolvidos individualmente; no entanto, pode discuti-los com os seus colegas e com os seus professores, antes de os resolver: a programação propriamente dita, essa sim, deve ser exclusivamente sua.
2. Se precisar de algum esclarecimento sobre os enunciados, ou sobre outros aspectos desta prova, peça-os por email ao Prof. Pedro Guerreiro, presidente do júri, para [pg@di.fct.unl.pt](mailto:pg@di.fct.unl.pt); as respostas virão por “reply”.
3. Entregue o seu trabalho numa disquete trancada, etiquetada com todos os elementos de identificação úteis; não entregue mais nada (além da carta que acompanha a disquete).
4. Na sua disquete devem vir todos os ficheiros fonte, os dois ficheiros executáveis, os dois ficheiros de texto referidos nos requisitos técnicos, e ainda um ficheiro de texto message.txt que você usará para indicar as suas coordenadas (nome, escola, telefone, email, etc.), para descrever o conteúdo da disquete (isto é, listar todos os ficheiros presentes) e para dizer ao júri o que lhe apetecer.
5. Não deve haver na disquete mais nada para além dos ficheiros indicados no número anterior.
6. Verifique cuidadosamente que a sua disquete não tem vírus: se tiver, será destruída e a prova desclassificada.
7. O júri poderá querer recompilar o seu programa; se a compilação falhar, a prova será desclassificada.
8. O júri apreciará o texto do seu programa: seja cuidadoso, claro, asseado, económico e use o seu melhor estilo de programação.
9. Se o seu programa acabar por ser muito parecido com os dos seus colegas que também concorrem, diga isso claramente, (no ficheiro message.txt), explicando porquê; caso contrário a sua prova pode ser desclassificada.