

2ª Lista de Exercícios

Turma	Data de Entrega
LP- Noturno	29/04/2018

Natureza do Trabalho: em dupla

Entregar a lista através do e-mail palomar.cris@gmail.com até às 23:59 do dia 29/04/2018. Enviar apenas os programas fonte (.c). Os nomes dos arquivos fontes deverão ter no máximo 8 caracteres (além da extensão) e não poderão conter caracteres especiais ou acentuação.

- ➔ Cada programa deverá conter um comentário com o Ra e nome do(s) aluno(s).
- ➔ Não serão aceitas listas entregues fora do prazo determinado
- ➔ Informar se compilado no Cygwin ou CodeBlocks
- ➔ A lista merecerá uma nota de 0 a 10, sendo o critério de avaliação definido pelo professor.
- ➔ Exercícios que sejam considerados como fruto de algum tipo de fraude por parte da equipe (como cópia total ou parcial, por exemplo) atribuirão nota zero na atividade para TODOS os envolvidos.
- ➔ IDENTAR todos os programas.
- ➔ Todos os programas deverão usar **NOTAÇÃO PONTEIRO**

EXERCÍCIOS

1. O Centro Paula Souza vem passando por significativas modificações nos últimos anos e uma delas se refere ao software para controle acadêmico que foi implantado de forma unificada para todas as Fatecs. O SIGA (esse é o nome do sistema de gestão) padronizou o registro acadêmico em todas as unidades, além de oferecer uma série de recursos muito interessantes para os professores, coordenadores de curso e secretários acadêmicos. Uma das mudanças mais perceptíveis proporcionadas pela implantação do SIGA foi a mudança no formato dos números de matrícula dos alunos, os chamados RAs. Antigamente esse dado variava grandemente entre as unidades, agora todas usam o mesmo formato, de 13 dígitos numéricos, na forma FFFCCCAASTNNN, onde:

FFF = Código numérico da Fatec (Sorocaba é 003, São Paulo é 002, etc)

CCC = Código do curso

048 – Análise e Desenvolvimento de Sistemas

061 – Sistemas Biomédicos

073 – Eletrônica Automotiva

074 – Logística

080 – Polímeros

081 – Processos Metalúrgicos

099 – Projetos Mecânicos

100 – Fabricação Mecânica

AA = Ano de ingresso do aluno

S = Semestre de ingresso do aluno (1=primeiro semestre, 2= segundo semestre)

T = Turno do curso (1=manhã/matutino, 2=tarde/vespertino, 3=noite/noturno)

NNN = Número sequencial do aluno

Por exemplo, o novo RA 0030481321099 significa

Fatec:	003 (Sorocaba)
Curso:	048 (Análise e Desenvolvimento de Sistemas)
Ano de ingresso:	13 (2013)
Semestre de ingresso:	2 (Segundo semestre)
Turno:	1 (manhã)
Número sequencial:	099

Na Fatec de Sorocaba, por exemplo, o RA antigo era composto por 8 caracteres, na forma CCAASNNN, onde:

CC = Sigla do curso

AD – Análise e Desenvolvimento de Sistemas (manhã)

AN – Análise e Desenvolvimento de Sistemas (noite)

SD – Sistemas Biomédicos (manhã)

LT – Logística (tarde)

PL – Polímeros (tarde)

PD – Projetos Mecânicos (manhã)

PN – Projetos Mecânicos (noite)

OD – Fabricação Mecânica (manhã)

ON – Fabricação Mecânica (noite)

AA = Ano de ingresso do aluno

S = Semestre de ingresso do aluno (1=primeiro semestre, 2= segundo semestre)

NNN = Número sequencial do aluno

Você foi designado para gerar um programa que, para um RA antigo da Fatec de Sorocaba informado, gere uma versão correspondente no novo formato de 13 caracteres, aproveitando nos últimos três dígitos a informação correspondente do RA antigo. Por exemplo, o RA antigo AD132099 seria o equivalente ao RA novo 0030481321099.

Entrada

Inicialmente um valor N é informado, indicando a quantidade de casos de teste a serem processados. Seguem-se N linhas, cada uma contendo uma string de 8 caracteres representando um RA antigo válido da Fatec de Sorocaba. Não serão fornecidos RAs antigos dos cursos de Eletrônica Automotiva e de Processos Metalúrgicos pois são cursos muito recentes e já foram implantados com os novos RAs.

Saída

Para cada caso de teste imprima o novo RA de 13 caracteres correspondente ao RA antigo lido.

Exemplo

Entrada:	Saída:
9	0030481011001
AD101001	0030481223035
AN122035	0030611111111
SD111111	0030741212221
LT121221	0030800912252
PL091252	0030990821124
PD082124	0030990713255
PN071255	0031001211022
OD121022	0031001213212
ON121212	

2. Escrever uma função que:

- receba duas strings como parâmetro, bem como um valor inteiro representando uma posição.
- insira a segunda string na primeira, na posição indicada pelo valor.
- Fazer um programa que receba duas strings do usuário, o valor da posição, chame a função anteriormente implementada e exiba o resultado ao usuário.
- Repetir o programa enquanto a posição indicada seja maior que zero.

Exemplo: S1: "Programacao"

S2: "Estruturada"

Posição: 9

Resultado: ProgramaEstruturadacao

3. Em uma dada string devemos encontrar o CPF do cliente e o valor de suas faturas pendentes, mas esses valores estão embaralhados em caracteres não numéricos. Sua tarefa é encontrar o CPF e somar o valor total das faturas pendentes de cada cliente.

Entrada:

A entrada é composta por um número que indica a quantidade de clientes. Cada cliente possui uma string contendo entre 1 e 60 caracteres ('0'-'9' e outros caracteres não numéricos). Os 11 primeiros dígitos numéricos correspondem ao CPF do cliente. Os demais números são separados pela casa decimal, que sempre vem acompanhada de um "." com dois dígitos na sequência indicando os centavos.

Saída:

Imprima o CPF do cliente, conforme apresentado abaixo e a soma dos valores das faturas em aberto, sempre com duas casas decimais.

Exemplo de Entrada:

```
3
7a5a6adfg4a4adsfgsd9fg6aa904aa91#$$%11.2342**&774667
11#$$%11111*&"1111&*(1198-=&8789."34"8%(&567.22%99
123ccc456dd875tt.q10120.891000.40
```

Exemplo de Saída

```
CPF: 75644969049 Soma: 42774779.23
CPF: 11111111111 Soma: 11997455.56
CPF: 12345687510 Soma: 1121.29
```

4. (3,0) Faça um jogo no qual o computador desafia dois usuários para ver quem tem mais memória. O jogo deve gerar um número de 0 a 9 e apresentar ao jogador 1 durante alguns segundos. Depois deve repetir este procedimento para o jogador 2. Em seguida, ele deve pedir ao jogador 1 para informar o número escolhido, e após pedir ao jogador 2 para fazer a mesma coisa. Na segunda rodada, o computador deve gerar mais um número entre 0 e 9, que será concatenado à direita do primeiro número. Então repetiremos o procedimento de exibição e teste de memorização para os dois usuários. O programa deve continuar adicionando números (casas adicionadas ao final do número) até que um dos dois jogadores não consiga mais memorizar perfeitamente o número. Para concluir, o programa deve informar qual dos dois jogadores ganhou, quantas casas foram memorizadas ao total, e em que casa foi que o jogador perdedor errou (primeira casa errada começando da esquerda para a direita).

Dicas: Procure desenvolver o programa em módulos(funções);
Armazene a sequência de números dentro de uma string.

Usar notação ponteiro.

Para gerar número randômico:

```
srand(time(NULL));
rand() % 9;
```