

6.0  
B



Universidade do Sul de Santa Catarina – UNISUL  
Curso de Ciência da Computação  
Unidade de Aprendizagem: Algoritmos II  
Professor: Clávison Martinelli Zapelini  
AVALIAÇÃO TEÓRICA I

E-mail: clavison.zapelini@unisul.br

Aluno(a): Tiago Buling

**Observações:**

- A avaliação se encerrará exatamente às 22:00 (sem intervalo). Deve ser executada de forma individual e sem consulta.
- As dúvidas em relação às questões serão esclarecidas somente nos momentos iniciais juntamente com a leitura da prova.
- Somente serão corrigidas as respostas que estiverem escritas à caneta e legíveis
- O valor de cada questão é de 2.5 pontos

1 - Em um supermercado, os produtos são organizados em categorias de acordo com o valor, conforme tabela a baixo.

Valor

Idade	Categoria
1,00 à 20,00	Classe D
21,00 à 50,00	Classe C
51,00 à 100,00	Classe B
Acima de 100,00	Classe A

Construa um algoritmo para cadastrar nome e valor de 150 produtos (somente o nome e o valor serão armazenados). Após o cadastro, permitir que o usuário digite um nome e o algoritmo retorne em qual categoria está o produto. OBS: O usuário poderá solicitar a categoria de quantos produtos desejar (utilizar estruturas de repetição que possibilite essa funcionalidade).

2 - Faça um algoritmo que crie um cadastro para o sexo e curso de 5.000 alunos. O curso será cadastrado como um booleano (verdadeiro o aluno cursa ciência da computação, falso o aluno cursa outro curso). Após a leitura, conte quantas alunas do sexo feminino estão matriculadas no curso de ciência da computação. Também, o percentual de alunos que cursam ciência da computação. Mostre o resultado na tela.

3 - Faça um algoritmo para cadastro de nomes e CPFs de N pessoas (no máximo 100 pessoas) e que apresente o seguinte menu (com as respectivas funcionalidades):

- 1 – Cadastrar pessoa
- 2 – Buscar pessoa por nome
- 3 – Buscar pessoa por CPF
- 4 – Sair

Para o cadastro (opção 1), solicitar o nome e o CPF cuidando do fato que não pode exceder 100 cadastros. Para as buscas (opções 2 e 3) deve mostrar o nome e o CPF de acordo com o filtro escolhido. Caso não exista o cadastro buscado, deve ser apresentada uma mensagem indicando a não existência desse cadastro.

SE ( $\epsilon_{PCA} = -1$ ) ENTÃO

Continuación

5

cont ← 0;  
 REPLITA NOME (FIMT = 0) → "F" = 0 → (S0)  
 escreva ("Nome"); → passat-met  
 leia (nomes[cont]);  
 escreval ("CPF"); → O → sat-000  
 leia (cptfs[cont]); → JAF = 0 → F0 → (S0)  
 (1 + sat-000 → sat-000)

cont ← cont + 1; ?  
 ENQUANTO (continuar = "S" e cont <= 100) FIM  
 //busca nomes

SE (opcao = 2) ENTÃO 0002 → (2) sat  
 escreva ("Busca por nome"); → sat-000  
 leia (busca); → \* (2) - sat → latr000  
 PARA (i = 0; i <= cont; i++) FAÇA  
 SE (busca = nomes[i]) ENTÃO  
 escreval ("Nomes", nomes[i]); → 0021AM  
 FIMSE

FIMPARA  $\Sigma$  não encontro? → antigoIA (E)

FIMSE

Flight Beijing → Doha & 265 Doha 32

# 1) Algoritmo "Supermercado" EWM17

$$\partial_1 \text{Var}(\mathbb{E}(f)) > \mathbb{E}[\partial_1 f] \text{ and } \partial_2^2 \leq \mathbb{E}[\partial_2^2 f].$$

nome: vetor[150] caractere;

Valor : vetor [ 150 ] float ;  
continuar, i : inteiro  
busca : caractere

**INICIO** DATA/E (opt) < FD 2d(SV) E;

interno ← 0; ("A B C D") nenhuma

PARA ( $i = 0$ ;  $i < 150$ ;  $i++$ ) FAÇA S2M17

```
escreva("Nome:");
```

leia(nome[i]);

escreva ("Valor"); ("") = word);  
leia (valor);

leia(valorLi1);  
i < i+1;

escreva ("continua") também

exclu<sup>s</sup> (continuar): p<sup>re</sup>sisim ou U p<sup>re</sup>leig (continuar) aib<sup>o</sup> "contar/pela"

Final continuous function  $\lim_{x \rightarrow \infty} f(x) = 200$

SE (continuar = 0) ENTÃO

~~inglés) ELM PARA (notar : 0276)~~

~~cost of M/S Emt, cost of m/s, cost of net, i  
M PADA~~

M PARK (bott : bouthwaite)

REPITA (Nenadov, 1971) OPIV

escreva ("Produto");

(eia) busca); + i; 0002 > i; 0 = i) ARAA

~~Self-busca~~ (100, 177) Tareas

~~SE busca = nombre EN TABLA~~

~~SE (valor1) = 1 el valor 5=20 EN  
PSCFEN ("CLASSE D")~~

~~External~~ (IMSE) ~~is~~

SE (valor[7] > 20 e valor[7] < 51) ENTÃO  
escreva ("CLASSE C");

FIMSE

// classe B

SE (valor[7] > 50 e valor[7] < 101) ENTÃO  
escreva ("CLASSE B");

FIMSE

// classe A

SE (valor[7] > 100) ENTÃO

FIMSE

FIMSE

ATÉ (busca = "")

FIMALGORITMO

2) Algoritmo "dois" (variáveis)

var

sexo: vetor[5000] caractere;

curso: vetor[5000] boolean;

i, fem-faz CCP, nao faz, faz CCP: inteiro;

percentual: real;

INÍCIO

i ← 0;

PARA (i = 0; i < 5000; i++) FAÇA

escreva ("CURSO CCP?");

leia (curso[i]);

escreva ("Sexo?");

leia (sexo[i]);

continuação  
da

(62)

Repetição

(63)

Repetição

(64)

Repetição

(65)

Repetição

(66)

Repetição

(67)

Repetição

(68)

Repetição

(69)

Repetição

(70)

Repetição

(71)

Repetição

(72)

Repetição

(73)

Repetição

(74)

Repetição

(75)

Repetição

(76)

Repetição

(77)

Repetição

(78)

Repetição

(79)

Repetição

(80)

Repetição

(81)

Repetição

(82)

Repetição

(83)

Repetição

(84)

Repetição

(85)

Repetição

(86)

Repetição

(87)

Repetição

(88)

Repetição

(89)

Repetição

(90)

Repetição

(91)

Repetição

(92)

Repetição

(93)

Repetição

(94)

Repetição

(95)

Repetição

(96)

Repetição

(97)

Repetição

(98)

Repetição

(99)

Repetição

(100)

Repetição

(101)

Repetição

(102)

Repetição

(103)

Repetição

(104)

Repetição

(105)

Repetição

(106)

Repetição

(107)

Repetição

(108)

Repetição

(109)

Repetição

(110)

Repetição

(111)

Repetição

(112)

Repetição

(113)

Repetição

(114)

Repetição

(115)

Repetição

(116)

Repetição

(117)

Repetição

(118)

Repetição

(119)

Repetição

(120)

Repetição

(121)

Repetição

(122)

Repetição

(123)

Repetição

(124)

Repetição

(125)

Repetição

(126)

Repetição

(127)

Repetição

(128)

Repetição

(129)

Repetição

(130)

Repetição

(131)

Repetição

(132)

Repetição

(133)

Repetição

(134)

Repetição

(135)

Repetição

(136)

Repetição

(137)

Repetição

(138)

Repetição

(139)

Repetição

(140)

Repetição

(141)

Repetição

(142)

Repetição

(143)

Repetição

(144)

Repetição

(145)

Repetição

(146)

Repetição

(147)

Repetição

(148)

Repetição

(149)

Repetição

(150)

Repetição