Algoritmos e Programação Estruturada
Avaliação N1

Nome:	
Matrícula:	
Data:	

A interpretação das questões faz parte da avaliação.

1 (0,1 pts) Qual das seguintes implementações do `for` está correta, conforme C89?

```c
int vetor[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

( )
```c
for (int i = 0; i < 9; i++) {
        printf("%d ", vetor[i]);
}
```

( )
```c
for (i = 0; i < 9; i++) {
        printf("%d ", vetor[i]);
}
```

( )
```c
for (int i = 0; i < 9; i++) {
        printf("%d ", &vetor[i]);
}
```

( )
```c
for (i = 0; i < 9, i++) {
        printf("%d ", vetor[i]);
}
```

2 (0,2 pts) Qual a saída?

```c
#include <stdio.h>

int main() {
        int i = 20;

        while (i < 10) {
                printf("while: %d\n", i);
                i++;
        }

        do {
                printf("do: %d\n", i);
                i++;
        } while (i < 10);

        return 0;
}
```

( ) `while: 20`

( ) `while: 21`

( ) `do: 20`

( ) `do: 21`

( ) Nenhuma das alternativas anteriores.

3 (0,2 pts) O Git é extremamente importante, especialmente no desenvolvimento de software. Quais as alternativas que são vantagens do GIT:

Marque **V** para as **verdadeiras** e **F** para as **falsas**.

( ) **Controle de Versão:** Git é um sistema de controle de versão que permite rastrear mudanças no código ao longo do tempo. Isso é crucial para poder voltar a versões anteriores do código se algo der errado.

( ) **Colaboração:** Com Git, múltiplos desenvolvedores podem trabalhar no mesmo projeto simultaneamente sem sobrescrever o trabalho uns dos outros. Isso facilita a colaboração em equipe e a integração de diferentes partes do projeto.

( ) **Descentralização:** Diferente de outros sistemas de controle de versão, Git é descentralizado. Cada desenvolvedor tem uma cópia completa do repositório, o que melhora a performance e permite trabalhar offline.

( ) **Segurança:** Git mantém um histórico completo e seguro de todas as mudanças feitas no código, o que ajuda a prevenir perda de dados e facilita a auditoria.

( ) **Flexibilidade:** Git permite criar e gerenciar diferentes branches (ramificações) do projeto, facilitando o desenvolvimento de novas funcionalidades e a correção de bugs sem afetar o código principal.

4 (0,1 pts) Qual a saída do seguinte código. Marque TODAS as alternativas possíveis.

```c
#include <stdio.h>
#include <string.h>

int main() {
        char boasvindas[] = "Bem-vindo, ";
        char nome[101];
        int i;

        printf("Digite seu nome: ");
        fgets(nome, 100, stdin);

        for (i = 0; i < strlen(nome); i++) {
                if (nome[i] == ' ') break;
        }

        strncat(boasvindas, nome, i);

        printf("%s", boasvindas);

        return 0;
}
```

Com a seguinte entrada:

```
Fulano de Tal
```

( ) `Error`

( ) `Bem-vindo, Fulano`

( ) `Bem-vindo, Fulano de Tal`

( ) `Bem-vindo,`

5 (0,1 pts) Quais formas abaixo permite a troca dos valores entre as variáveis?

Sejam duas variáveis:

```c
int a;
int b;
```

Com valores inseridos pelo usuário via teclado.
Quais formas abaixo permite a troca dos valores entre as variáveis? (Selecione todas as possíveis)

( )
```c
a = b;
b = a;
```

( ) Não é possível

( )
```
aux = a;
a = b;
b = aux;
```

( )
```
a = a + b;
b = a - b;
a = a - b;
```

( )
```
aux = a;
b = a;
a = aux;
```

6 (0,2 pts) Qual a saída do seguinte código?

```c
#include <stdio.h>

int main() {
    int a;
    int b;
    float c;

    scanf("%d %d", &a, &b);

    if(a > b || !(a > 0)) {
        c = (float)(b / a);
    } else {
        c = (float)(a / b);
    }
    printf("%.2f\n", c);
    return 0;
}

```

Com a seguinte entrada:

```
5
11
```

( ) 0.00

( ) 0.45

( ) 2.00

( ) 2.20

( ) Nenhuma das alternativas anteriores.

7 (0,1 pts) Qual trecho de código abaixo, em C89, gera a seguinte saída:

```
*
**
***
****
```

para $n = 4$.

( )
```
for(linha = 1; linha <= n; linha++) {
    for (coluna = 1; coluna <= linha; coluna++) {
        printf("*");
    }
    printf("\n");
}
```

( )
```
for(linha = 1; linha <= n; linha++) {
    for (coluna = 1; coluna < linha; coluna++) {
        printf("*");
    }
    printf("\n");
}
```

( )
```
for(linha = 1; linha <= n; linha++) {
    for (coluna = linha; coluna <= 4; coluna++) {
                printf("*");
    }
    printf("\n");
}
```

8 (0,1 pts) Em C89, não existe um tipo primitivo string. Mas string existe em C, em textos envolvidos por aspas duplas.

( ) Verdadeiro

( ) Falso

9 (0,1 pts) Qual o valor de `d` após executar o seguinte trecho código:

```
a = 1;
b = 2;
c = 3;
d = ++a * b - c--;
```

Qual o valor de **d**:

( ) 0

( ) 1

( ) 2

( ) 3

( ) 4

10 (0,1 pts) Qual o valor de `w` após a execução do seguinte trecho código:

```
y = 5;
z = 11;
w = y + z;
if (y > z)  {
    w = y * z;
}
```

Qual o valor de **w** após a execução do seguinte código:

( ) 0

( ) 5

( ) 11

( ) 16

( ) 55

11 (0,1 pts) Qual das seguintes implementações garante que um estudante é aprovado se ele obtiver nota mínima de 7 e frequência mínima de 75%:

( )
```
if (nota >= 7 && frequencia >= 0.75) {
    printf("aprovado");
}
```

( )
```
if (nota > 7 && frequencia >= 0.75) {
    printf("aprovado");
}
```

( )
```c
if (nota >= 7 || frequencia >= 0.75) {
    printf("aprovado");
}
```

( )
```c
if (nota > 7 && frequencia > 0.75) {
    printf("aprovado");
}
```

( )
```c
if (nota > 7 || frequencia > 0.75) {
    printf("aprovado");
}
```

12 (0,1 pts) Qual a saída do seguinte trecho de código:

```c
int vet1[] = {1, 2, 3};
int vet2[] = {1, 2, 3};
if (vet1 == vet2) {
    printf("São iguais");
} else {
    printf("Não são iguais");
}
```

( ) São iguais

( ) Não são iguais

( ) Ocorrerá um erro.

( ) Não irá exibir nada.