



Aluno: Ricardo Anchieta Junior

Prova 2

Tipo 2

Questão 1:

(1,0 pt) Qual será a saída do seguinte programa em C?

```
int main(){
    int i;
    int *ptr = (int *) malloc(5 * sizeof(int));

    for (i=0; i<5; i++)
        *(ptr + i) = i;

    printf("%d ", *ptr++);
    printf("%d ", (*ptr)++);
    printf("%d ", *ptr);
    printf("%d ", ++*ptr);
    printf("%d ", ++*ptr);
}
```

- a) Erro de compilação
b) 0 1 2 2 3

~~c) 0 1 2 3 4
d) 1 2 3 4 5~~

e) N.D.A.

Questão 2:

```
typedef struct {
    char nome[100];
    int telefone;
} Pessoa;
```

Crie um programa em C para o cadastro de pessoas a partir do teclado em um vetor dinâmico (**Pessoa*** agenda), inicialmente de tamanho igual a zero. Cada vez que uma nova pessoa for introduzida, o bloco de memória apontado por agenda é aumentado pelo tamanho de uma Pessoa. Para encerrar o cadastro e sair do programa, o usuário deverá entrar com um n° de telefone negativo.

Dica: utilize as funções void* malloc(unsigned size) e void* realloc(void* ptr, unsigned size)

2º

```
VOID ADDPESSOA (PESSOA *AGENDA, INT INDICE, INT NUMERO, CHAR NOMES)
```

```
{  
    IF (*AGENDA == NULL) {
```

```
        PRINTF("ERRO");
```

```
        RETURN -1;
```

```
    }  
}
```

```
STRCPY(AGENDA[INDICE], AGENDA + INDICE, NOMES);
```

```
AGENDA + INDICE -> TELEFONE = NUMERO;
```

```
}
```

```
INT MAIN() {
```

```
    PESSOA *AGENDA;
```

```
    INT TAM, NUMERO, IND, RESP;
```

```
    CHAR NOMES;
```

```
    IND = 0;
```

```
    AGENDA = (PESSOA*) MALLOC(SIZEOF(PESSOA));
```

```
    PRINTF("DESEJA ADICIONAR ALGUEM A AGENDA? 1-SIM, 0-NÃO");
```

```
    SCANF("%d", &RESP);
```

```
    WHILE (RESP == 1) {
```

```
        AGENDA = REALLOC(AGENDA, SIZEOF(PESSOA) * (TAM + 1));
```

```
        PRINTF("INSIRA O NOME E O TELEFONE DA PESSOA");
```

```
        SCANF("%s %d", NOMES, &NUMERO);
```

```
        IF (NUMERO < 0) {
```

```
            RETURN 0;
```

```
            ADDPESSOA (AGENDA, IND, NUMERO, NOMES);
```

```
            FOR (INT I = 0; I < TAM; I++) {
```

```
                PRINTF("%s %d", AGENDA + I, AGENDA + I + 1);
```

```
            RETURN 0;
```

```
            IND++;
```


Ricardo Anchieta

Questão 3:

Qual será a saída do seguinte programa em C?

```
#include <stdio.h>
int main()
{
    char *ptr = "helloworld";
    printf("%s", ++ptr + 3);
    return 0;
}
```

2,5

~~a) oworld~~

b) world

c) hell

d) hello

Questão 4:

(3,0 pts) O código de César é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de cifra de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes (k). Considera-se a lista de alfabeto como sendo circular. Por ex.: com $k = 3$, 'A' seria substituído por 'D', 'Z' se tornaria 'C', e assim por diante.

Utilizando o código de César, crie um programa em C que leia senhas criptografadas de um banco de dados (bd.txt) contendo senhas de até n caracteres, e as mostre descriptografadas na tela.

Ex. de arquivo de entrada:

uhsjsm1
uhsjsm2
uhsjsm3

Saída na tela:

senha1
senha2
senha3

(112)

CHAR DESCRIPTOGRAFAR (CHAR S1, CHAR S2, INT K) {

INT CONT = 0

WHILE (S1[CONT] != '\0') {

S2[CONT] = S1[CONT] - K;

IF (S1[CONT] - K) > 122 {

S2[CONT] = S2[CONT] - 26

}

IF (S1[CONT] - K) < 97 {

S2[CONT] = S2[CONT] + 26;

}

CONT++;

RETURN S2;

}

INT MAIN() {

FILE *ARQ;

INT K;

CHAR STR1[100], STR2[100];

ARQ = FOPEN("BD.TXT", "R");

IF (ARQ == NULL) {

PRINTF("ERRO DE ABERTURA");

RETURN -1;

}

SCANF("%d", &K); // USUARIO COLOCA VALOR DE K

WHILE (FGETS(STR1, 100, ARQ) != NULL) {

DESCRIPTOGRAFAR(STR1, STR2, K);

RETORNO;

}, fultou me deu na tela