

Nome: _____ **GABARITO** _____

Obs.: A prova pode ser feita a lápis! As funções não devem conter valores fixos!

1. O que será impresso na tela? Mostre o valor que cada variável assume durante a execução e também circule a resposta; não é para explicar o que cada linha faz.

a) (1 ponto)

```
main() {
    char str[]="um dois três";
    char aux[13];
    int tam=strlen(str);
    int i;
    for (i=tam-1; i>0; i--)
        if (str[i]==' ') {
            strcpy(aux, &str[i+1]);
            break;
        }
    printf("%s", aux);
}
```

três

b) (1 ponto)

```
void funcao(int *y) {
    int i=0;
    for (;i<3;i++)
        *y= *y * 2;
}
```

```
main(void) {
    int x=2;
    funcao(&x);
    printf("%d",x);
}
```

16

c) (1 ponto)

```
typedef struct {
    int x,y;
} Ponto;
```

```
funcao(Ponto *p3) {
    p3->x = p3->x + 2;
    p3->y = p3->y + 2;
}
```

```
main(void) {
    Ponto *p1, p2;
    p2.x = 2;   p2.y=4;
    p1 = &p2;
    funcao(p1);
    printf("%d", p2.x + p2.y);
}
```

10

d) (1 ponto)

```
main() {
    int i,j,valor=1;
    int vet[3][3];
    for(i=0;i<3;i++) {
        for (j=0;j<3;j++) {
            vet[i][j]=valor;
            valor++;
        }
    }
    printf("\n%d",vet[2][1]);
}
```

8

e) (1 ponto)

```
main() {
    int tam;
    char *p1,*p2;
    p1 = (char *)malloc(25);
    p2 = (char *)malloc(25);
    strcpy(p1,"frase");
    tam=strlen(p1);
    p1[tam]='.';
    p1[tam+1]='\0';
    strcpy(p2,"teste");
    strcat(p1,p2);
    printf("%s",p1);
}

// char *strcat(char *destino, char *origem);
```

frase.teste

2. (2 pontos) Desenvolva uma função denominada preencheLetras() que recebe o tamanho de um novo vetor que será retornado pela função. O novo vetor deve ser preenchido com letras aleatórias.

```
main() {
    int i,tam=10;
    char *vetor;
    srand(time(NULL));
    vetor = preencheLetras(tam);
    printf("\nVetor Aleatório:\n");
    for (i=0; i<tam; i++)
        printf("%c ", vetor[i]);
}
```

```
char * preencheLetras(int tam) {
    int i;
    char *vet = (char *) malloc(tam*sizeof(char)+1);
    for (i=0; i<tam; i++) {
        vet[i]='a'+(rand()%26);
    }
    vet[i]='\0';
    return vet;
}
```

```
main() {
    int i,tam=10;
    char *vetor;
    srand(time(NULL));
    vetor = preencheLetras(tam);
}
```

```

printf("\nVetor Aleatório:\n");
for (i=0; i<tam; i++)
    printf("%c ", vetor[i]);
}

```

3. Tendo o código abaixo.

```

#include <stdio.h>

typedef struct funcionario {
    char nome[50];
    float salario;
    char sexo;
} Funcionario;

void imprimeCadastro(Funcionario *f, int n) {
    int i=0;
    for(;i<n;i++) {
        imprimeEstrutura(&f[i]);
    // ou:
        printf("\nNome=%s", f[i].nome);
    //     printf("\nsalário=%.2f", f[i].salario);
    //     printf("\nsexo=%c\n", f[i].sexo);
    }
}

main() {
    Funcionario f1, *f2, f3[3]={{ "joao", 1000.0, 'm' },
                                { "maria", 2000.5, 'f' },
                                { "pedro", 500.5, 'm' }};

    f2 = alocaFuncionario("Fulano de tal", 2000.0, 'm');
    imprimeEstrutura(f2);
    imprimeCadastro(f3,3);
    gravaNomesArquivo("nomes.txt", f3, 3);
}

```

3a. (1 ponto) Implemente a função alocaFuncionario().

3b. (1 ponto) Implemente a função imprimeEstrutura, a qual recebe um ponteiro para uma estrutura e mostra todos os seus campos.

3c. (1 ponto) Implemente a função gravaNomesArquivo(), a qual grava todos os nomes do vetor de estrutura, um em cada linha do arquivo.

```

#include <stdio.h>

```

```

typedef struct funcionario {
    char nome[50];
    float salario;
    char sexo;
} Funcionario;

```

```

Funcionario * alocaFuncionario(char *nome, float salario, char sexo) {
    Funcionario *f = (Funcionario *) malloc(sizeof(Funcionario));
    strcpy(f->nome, nome);
    f->salario = salario;
    f->sexo = sexo;
    return f;
}

```

```

void imprimeEstrutura(Funcionario *f) {
    printf("\nNome=%s", f->nome);
    printf("\nsalário=%.2f", f->salario);
    printf("\nsexo=%c\n", f->sexo);
}

```

```

void imprimeCadastro(Funcionario *f, int n) {
    int i=0;
    for(;i<n;i++) {
        imprimeEstrutura(&f[i]);
// ou:    printf("\nNome=%s", f[i].nome);
//        printf("\nsalário=%.2f", f[i].salario);
//        printf("\nsexo=%c\n", f[i].sexo);
    }
}

void gravaNomesArquivo(char *nomeArq, Funcionario *f, int n) {
    FILE *fp;
    int i;
    fp = fopen(nomeArq, "w");
    for (i=0; i<n; i++)
        fprintf(fp, "%s\n",f[i].nome);
    fclose(fp);
}

main() {
    Funcionario f1, *f2, f3[3]={{'joao', 1000.0, 'm'},
                               {'maria', 2000.5, 'f'},
                               {'pedro', 500.5, 'm'}};

    f2 = alocaFuncionario("Fulano de tal", 2000.0, 'm');
    imprimeEstrutura(f2);
    imprimeCadastro(f3,3);
    gravaNomesArquivo("nomes.txt", f3, 3);
}

```