

Nome: \_\_\_\_\_ **GABARITO** \_\_\_\_\_

*Obs.: A prova pode ser feita a lápis!*

**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)** (1pt)

```
#include <stdio.h>
#include <stdlib.h>

#define MAX 5

main(void) {
    int *x,*y;
    int i=0;
    x = (int *) malloc(MAX*sizeof(int));
    for (y=x; i<MAX; i++,y++)
        *y = i + 1;
    for(i=0; i<MAX; i++)
        printf("%d ",x[i]);
}
```

**1 2 3 4 5**

**b)** (1pt)

```
main(){
    int vet[5]={1,2,3,4,5};
    int *p,i;
    p = &vet[4];

    for(i=4; i>=0; p=p-2, i=i-2)
        *p = *p + 2;

    for(i=0; i<5; i++)
        printf("%d ", vet[i]);
}
```

**3 2 5 4 7**

**c)** (1pt)

```
#include <string.h>

main(void) {
    char str[50]="abc";
    char *s=(char *)malloc(50*sizeof(char));
    int i, fim=strlen(str);
    strcpy(s, str);
    for (i=0;i<2;i++, fim++)
        s[fim] = str[i];
    s[fim]='\0';
    printf("%s", s);
}
```

**abcab**

**d)** (1pt)

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

int main ()
{
    char str[] ="Departamento de Computacao";
```

```

char * ptr;
ptr = strstr (str,"ta");
*ptr = 'o';
*(ptr+1)='\0';
printf("%s",str);
return 0;
}

```

## Deparo

**e)** (1pt)

```

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

int main ()
{
    char str[] ="Departamento de Computacao";
    char str2[10];
    char * ptr;
    ptr = strstr (str,"Comp");
    strncpy (str2,ptr,7);
    str2[7]='\0';
    puts (str2);
    return 0;
}

```

## Computa

**f)** (1,5pts)

```

#include <stdio.h>
#include <stdlib.h>

#define MAX 26

int funcao(char * vet, int n) {
    static int count = 0;
    if (!(count % 3))
        if (count < n)
            printf("%c ", vet[count]);
        else
            return 0;
    count++;
    return 1;
}

char * getVetor() {
    int i;
    char * v = (char *) malloc(MAX*sizeof(char));
    for (i=0; i<MAX; i++)
        v[i]=97+i;
    return v;
}

int main() {
    int i;
    char * vet=getVetor();
    for(i = 0; i < 10; i++)
        if (!funcao(vet, MAX)) break;
    return 0;
}

```

## a d g j

**2.** (3,5pts) Desenvolva uma função que conta o número de palavras (qtdePalavras) de uma string recebida como parâmetro e retorna um vetor com tamanho qtdePalavras, contendo o número de caracteres de cada palavra. Faça um teste com uma string "uma frase de exemplo" e imprima o vetor resultante na main() "3 5 2 7". Confira se a última palavra também entrou no vetor. Não pode ser usada variável global.

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

```
#include <string.h>
#include <stdlib.h>
```

```
int *contaPalavras(char *str, int *tamVetor) {
    int i=0, j, contPalavras=0, contLetras;
    int *vetor;

    for(;i<strlen(str);i++) {
        if (str[i]==' ')
            contPalavras++;
    }
    contPalavras++;
    *tamVetor = contPalavras;

    vetor = (int *) malloc(contPalavras*sizeof(int));

    for(i=0,contLetras=0,j=0;i<strlen(str);i++) {
        contLetras++;
        if (str[i]==' ') {
            vetor[j]=contLetras-1;
            contLetras=0;
            j++;
        }
    }
    vetor[j]=contLetras;

    return vetor;
}
```

```
main(void) {
    char frase[]="uma frase de exemplo";
    int tam,i;
    int *vetor = contaPalavras(frase,&tam);
    printf("\n%s\n",frase);
    for (i=0; i<tam;i++)
        printf("%d ",vetor[i]);
}
```