

AVALIAÇÃO AP2 N1 – 2025.1
DISCIPLINA DE ESTRUTURA DADOS

Vetores e alocação dinâmica, Matrizes e Cadeias de caracteres

Nome:

Data:

- 1) (3,0 pontos) Analisando o código abaixo, preencha o esquemático representativo da memória principal (tabela a direita). Imagine que o programa resultado da compilação do seguinte código em C está em execução na memória e que a função comprimento está prestes a retornar o valor “n” já computado para a função principal.

- Para efeito de simplificação, considere que os frames que compõe o *stack* são compostos apenas de variáveis locais às funções e que o espaço de endereçamento é contínuo na memória. Use `sizeof(char*)`: 8 Bytes, `sizeof(int)`: 4 Bytes e `sizeof(char)`: 1 Byte.

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

```
int comprimento(char* string){  
    int n = 0;  
    for(int i = 0; string[i] != '\0'; i++)  
        n++;  
    return n;  
}  
  
int main(void){  
    char palavra[] = "sol";  
    printf("%d", comprimento(palavra));  
    return 0;  
}
```

Endereço	Conteúdo
0x10000	Código Objeto
0x10200	
0x10204	
0x10208	
0x1020C	
0x10210	
0x10214	
0x10218	
0x1021C	
0x10220	
0x10224	

- 2) (3,0 pontos) Implemente uma função para, dada uma matriz alocada dinamicamente conforme visto em aula (com ponteiro de ponteiros), retornar a quantidade de ocorrências de números iguais a “num”, conforme protótipo:

```
int conta_num_em_mat(float** mat, float num);
```

- 3) (4,0 pontos) Implemente uma função em C que receba como parâmetro uma string terminada em '\0' e retorne um ponteiro para uma nova string, alocada dinamicamente, contendo uma cópia da original com todas as letras minúsculas convertidas para maiúsculas.

`char* uppercase_copy(char * string);`

Restrições:

Não utilize nenhuma função da biblioteca padrão, como as de `string.h` ou `ctype.h`.

A string original não deve ser alterada.