

Nome:

Data:

- 1) (3,0 pontos) Analise o código em C a seguir e complete o **esquema da memória principal** (tabela à direita). Considere que o programa já foi compilado e está em execução, e que a última atribuição ao vetor `vet` já foi realizada.

Para simplificação:

- Os frames da pilha (**stack**) contêm apenas variáveis locais às funções.
- O espaço de endereçamento da memória é **contíguo**.
- Considere os seguintes tamanhos:
 - `sizeof(int)` = 4 bytes
 - `sizeof(float)` = 4 bytes
 - `sizeof(float*)` = 8 bytes

```
int main(void){  
static char letra = 'a'; // 'a' == 97  
  
float* vet = (float*)malloc(3*sizeof(float));  
  
for(int i = 0; i < 3; i++)  
    vet[i] = i + letra;  
  
return 0;  
}
```

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

- 2) (3,0 pontos) Implemente a função:

```
float** matriz_soma(float** A, float** B, int linhas, int colunas);
```

Essa função deve alocar dinamicamente uma nova matriz de tamanho `linhas × colunas`, onde cada elemento seja a soma dos valores correspondentes em `A` e `B`. Em seguida, deve retornar o ponteiro para a matriz resultante.

- 3) (4,0 pontos) Implemente a função `char* duplica(const char* s)`, que deve alocar dinamicamente memória suficiente para armazenar uma cópia da string `s` (incluindo o caractere `'\0'`), copiar todo o conteúdo de `s` para essa nova área de memória e retornar o ponteiro para a string resultante.