



Data de divulgação: 18/09/2017

Alocação dinâmica de memória

Questão 1

Explique as funcionalidades das funções da biblioteca `stdlib`, indicando os parâmetros recebidos como argumento e o que elas retornam:

1. `sizeof`
2. `malloc`
3. `free`
4. `calloc`
5. `realloc`

Questão 2

Formule os comandos em linguagem C para alocar dinamicamente os blocos de dados

1. Vetor de 1024 Bytes (1 Kbyte).
2. Matriz de inteiros de dimensão 10×10 .
3. Vetor para armazenar 50 registros contendo: nome do produto (30 caracteres), código do produto (inteiro) e preço em reais.
4. Texto de até 100 linhas com até 80 caracteres em cada linha.

Questão 3

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int * f(int *vetor, int n)
5  {
6
7      int i,*out;
8
9      /* PARTE 3 */
10
11     for(i =0; i < n;i++)
12     {
13         out[i] = vetor[i]*vetor[i];
14     }
15
16     return out;
```

```

17 }
18 }
19
20 void imprime(int *input, int n)
21 {
22     /* Código -fonte para imprimir um vetor de inteiros*/
23 }
24
25 int main()
26 {
27     int n,*valores,*quadrado;
28
29     scanf("%d",&n);
30
31     /*PARTE 1*/
32
33     quadrado = f(valores,n);
34
35     imprime(quadrado,n);
36
37     /*PARTE 2*/
38
39     return 0;
40 }

```

1. Como verificar se uma alocação dinâmica foi bem sucedida?
2. Complete o código-fonte visando implementar as PARTES 1, 2, 3 e 4.
 - PARTE 1: Alocar dinamicamente o vetor valores e preenchê-lo com valores aleatórios
 - PARTE 2: Liberar o espaço em memória ocupado pelas variáveis valores e quadrado
 - PARTE 3: Verificar se vetor existe.
3. Implemente a função `imprime` utilizando apenas aritmética de ponteiros e verificando se o vetor `input` existe.

Questão 4

Com base no código-fonte abaixo, responda:

1. Existe algum erro no código-fonte?
2. Explique as diferenças no uso dos ponteiros inteiros `ptr1` e `ptr2`.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int *ptr1,*ptr2;
7      int i;
8
9      ptr1 = (int *) malloc (sizeof(int));
10     ptr2 = &i;
11     *ptr1 = i;
12
13     printf("%d %d\n",*ptr1,*ptr2);
14
15     free(ptr1);

```

```

16
17     return 0;
18 }

```

Questão 5

Escreva um programa em linguagem C que receba um valor n para alocar dinamicamente um vetor de ponto flutuante duplo (double), em que seus elementos são aleatoriamente preenchidos. Em seguida, reduza o tamanho desse vetor pela metade.

Questão 6

Responda se a execução do código-fonte abaixo ocorre normalmente, desconsiderando avisos no momento da compilação. Explique a razão pela qual não é indicado que um ponteiro aponte para uma variável de tipo diferente.

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int *p;
6      char ct = 'C';
7
8      p = &ct;
9
10     printf("%c\n",*p);
11
12     return 0;
13 }

```

Questão 7

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct reg
5  {
6      char *nome;    // String de tamanho 20
7      float *peso;   // Valor float único
8  };
9
10 typedef struct reg Registro;
11
12 int main()
13 {
14     Registro *pessoa;
15     float f;
16
17     /*
18     ... PARTE 1
19     Bloco de código para alocar dinamicamente a variavel pessoa
20     */
21
22     printf("Digite o primeiro nome: ");
23     scanf("%s",pessoa->nome);
24

```

```

25     printf("Digite o peso: ");
26     scanf("%f", pessoa->peso);
27
28     printf("%s e %f \n", pessoa->nome, *(pessoa->peso));
29
30     /*
31     ... PARTE 2
32     Bloco de código para liberar a memória referente às variáveis alocadas
33     */
34
35
36     return 0;
37 }

```

O programa acima apenas lê informações referentes a uma pessoa e as imprime na tela. No entanto, seu código-fonte está incompleto e pode apresentar erros. Implemente as partes 1 e 2 do código-fonte acima e identifique eventuais erros, visando manter seu funcionamento sem erros de compilação, execução e vazamentos de memória.

Questão 8

Elabore uma função em linguagem C que receba um vetor de ponteiros inteiros `vInt` com n posições. Assume-se que cada posição aponta para um vetor de inteiros alocado dinamicamente, sendo que cada um pode apresentar tamanhos diferentes. Como saída, a função deve retornar um vetor de inteiros alocado de maneira dinâmica, em que cada posição i é a média dos valores do vetor de inteiros i em `vInt`.