UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC
DCC302 – ESTRUTURA DE DADOS I
PROF. MSC. ACAUAN C. RIBEIRO

Nome(s): **Guilherme Lucas Pereira Bernardo**

Nota:_____ **EXERCÍCIO - Aula 05 - Recursão, Structs e Alocação Dinâmica**

Questão 1) Calcular a soma entre dois números n1 e n2 incluindo os limites Soma(3,7) = 3+4+5+6+7 = 25

R:

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

int soma(int n1, int n2) {
     int cont = abs(n2-n1);
     int som = n1-n2;
     if (cont == 0) return som;
     else if (cont == 1) return (som+n2);
     else if (cont >1) som = som + soma(n1+1,n2);
     return som;
}

int main(){
     printf("%d", soma(5,3));
     return 0;
}
```

Questão 2) Faça uma função que calcula a potência XY , sem a utilização dos operadores de potenciação.

R:

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

int power(int base, int exp){
     if (exp == 0) return 1;
     else if (exp == 1)  return base;
     else if (exp>1) base = base*power(base, exp-1);
}
```

```
int main(){
     int x = 2;
     int y = 3;
     printf ("\n %d \n", power(x, y));
}
```

Questão 3) Escreva um programa em C que manipule um vetor de inteiros não nulos alocado dinamicamente.
→ O programa recebe inteiros, através da entrada padrão, e os insere no vetor.
→ A cada inteiro que é inserido a área de memória necessária para armazenar um inteiro é incrementada ao número de bytes necessários para armazenar o vetor.
→ O vetor não ocupa memória inicialmente. Quando o usuário entrar com o inteiro 0 (zero), o programa será finalizado e o mesmo não pertencerá ao vetor.
→ Após o processo de inserção o vetor deve ser impresso na saída padrão. Libere a memória utilizada antes do final do processamento.
R:
Questão 4) Com base no que vimos, construa um programa que aloque dinamicamente memória para um vetor de strings.
→ O processamento se dará da seguinte forma: o usuário fornecerá através da entrada padrão um conjunto de strings com tamanhos aleatórios. → O final de uma string é identificado pelo 09-ppppppppp09o-ppppppressionamento da tecla enter e o final do conjunto de strings é identificado pelo fornecimento de uma string chamada "exit".
→ Ao final do processamento o programa deve retornar na saída padrão as strings contidas no vetor.
R:

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

int main(){
   int *ptr, i;
   ptr = (int*)malloc(5 * sizeof(int));

   for(i = 0 ; i < 5; i++){
      printf("Digite um numero: ", i+1);
      scanf("%d", &ptr[i]);
   }
   for(i = 0 ; i < 5; i++){
      printf("%d  ", ptr[i]);
   }
   free(ptr);
   return 0;
}
//ta errado a implementação mas o q vale eh a tentativa
```

Questão 5) Faça um programa em C que utilize structs para armazenar os dados de um funcionário de uma empresa. Um funcionário de uma empresa deve possuir:

- **Nome (string de até 30 caracteres)**
- **Idade**
- **Sexo (representado por um caractere, 'M' ou 'F')**
- **CPF (armazenado em string)**
- **Cargo que ocupa (string de até 30 caracteres)**
- **Salário**
- **Data de Nascimento (dia e ano números inteiros, mês deve ser uma string)**

Você pode definir quantas estruturas achar necessário. Seu programa deve criar um vetor de 3 funcionários. Use a diretiva define para definir o tamanho do vetor. Em seguida, o usuário deve entrar com as informações para preencher esse vetor. Finalmente, seu programa deve imprimir o vetor preenchido.

R:

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

#define TAM 3

typedef struct
{
 int dia;
 int ano;
 char mes[30];
} Data;

typedef struct
{
 char nome[31];
 int idade;
 char sexo;
 char cpf[12];
 char cargo[31];
 float salario;
 Data nascimento;
 int cod_setor;
} Funcionario;

int main(int argc, char *argv[])
{
 Funcionario funcionarios[TAM];

 // Pega as informacoes para preencher o vetor
 for (int i = 0; i < TAM; i++){
   printf("Entrada de dados do funcionario %d \n", i + 1);
   printf("\t Nome: ");
   scanf(" %c", &funcionarios[i].nome);
   printf("\t Idade: ");
   scanf(" %d", &funcionarios[i].idade);
   printf("\t Sexo: ");
   scanf(" %c", &funcionarios[i].sexo);
   printf("\t CPF: ");
   scanf(" %c", &funcionarios[i].cpf);
```

```
   printf("\t Cargo: ");
   scanf(" %c", &funcionarios[i].cargo);
   printf("\t Salario: ");
   scanf(" %f", &funcionarios[i].salario);
   printf("\t Data de nascimento: ");
       scanf("  %d  %c  %d",  &funcionarios[i].nascimento.dia,
&funcionarios[i].nascimento.mes,
&funcionarios[i].nascimento.ano);
   printf("\t Codigo do setor: ");
   scanf(" %d", &funcionarios[i].cod_setor);
 }

 // Imprime informações
 printf("\n \n Imprimindo vetor: \n \n");
 for (int i = 0; i < TAM; i++)
 {
   printf("Funcionario %d : \n", i + 1);
   printf("\t Nome: %s \n ", funcionarios[i].nome);
   printf("\t Idade : %d \n", funcionarios[i].idade);
   printf("\t Sexo: %c \n ", funcionarios[i].sexo);
   printf("\t CPF: %s \n", funcionarios[i].cpf);
   printf("\t Cargo: %s \n", funcionarios[i].cargo);
   printf("\t Salario: %.2f  \n", funcionarios[i].salario);
     printf("\t  Data  de  nascimento:  %d  de  %s  de  %d  \n",
funcionarios[i].nascimento.dia,
funcionarios[i].nascimento.mes,
funcionarios[i].nascimento.ano);
         printf("\t   Codigo   do   setor:   %d   \n   \n",
funcionarios[i].cod_setor);
 }

 return 0;
}
```