UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

PROCESSO SELETIVO – 004 / 2023

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 1: Dados dois inteiros, **n** e **k**, escreva um programa em Linguagem C que imprima na saída padrão os **n** menores números primos maiores que **k**. Por exemplo, para **n = 5** e **k = 10** o programa imprime **11, 13, 17, 19, 23**.

```c
#include <stdio.h>

// Verifica se um número é primo (50%)
int eh_primo( int n ){
   int i, cont = 0;
   for( i = 1 ; i <= n ; i++ )
      if( n % i == 0 ) // forma simples, contagem de divisores.
         cont++;
   return cont == 2;
}

int main(int argc, char *argv[]) {
   int n, k;
   printf("Quantos primos? ");
   scanf("%d", &n);
   printf("A partir de qual valor: ");
   scanf("%d", &k);
   // Gera a série de n primos acima de k (50%)
   int cont_p = 0, x = k+1;
   while( cont_p < n ){
      if( eh_primo( x ) ){
         cont_p++;
         printf("%d : %d\n", cont_p, x );
      }
      x++;
   }
   return 0;
}
```

Membros da Banca:

Ricardo Ferreira Martins	**Yuri Kaszubowski Lopes**
Carlos Norberto Vetorazzi Jr.	**Rui Jorge Tramontin Jr. (Presidente)**

UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

PROCESSO SELETIVO – 005 / 2022

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 2: Na teoria de Sistemas, define-se elemento *minimax* de uma matriz como o menor elemento da linha em que se encontra o maior elemento da matriz. Escreva um algoritmo que, dada uma matriz A NxM (onde N e M são valores constantes), determine o elemento minimax desta matriz, escrevendo na tela o seu valor e suas coordenadas (linha, coluna).

```
#include<stdio.h>
#define N 5
#define M 5
int main(){
   // Obs.: esta resposta considera uma matriz do tipo int.
   int matriz[N][M];
   int i, j;
   printf("\nEntre com os valores da matriz: ");
   for( i = 0 ; i < N ; i++ ) {
      printf("\n Linha %d\n", i);
      for( j = 0 ; j < M ; j++ ) {
         printf("(%d)\t", j );
         scanf("%d", &matriz[i][j]);
      }
   }
   // Encontra o maior valor e a linha onde se encontra (i_maior). (40%)
   int i_maior = 0, maior = matriz[0][0];
   for( i = 0 ; i < N ; i++ )
      for( j = 0 ; j < M ; j++ )
         if( matriz[i][j] > maior ){
            i_maior = i;
            maior = matriz[i][j];
         }
   // Procura o menor na linha do maior --> minimax. (40%)
   int j_menor = 0, minimax = matriz[i_maior][0];
   for( j = 0 ; j < M ; j++ )
      if( matriz[i_maior][j] < minimax ){
         j_menor = j;
         minimax = matriz[i_maior][j];
      }
   // Saída dos dados. (incluído nos 20% da entrada)
   printf("Minimax: %d, nas coordenadas (%d, %d)\n", minimax, i_maior, j_menor);
   return 0;
}
```

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. Algoritmos: lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014. Capítulos 7 e 8
SOUZA, Marco Antonio Furlan de et al. Algoritmos e lógica de programação. São Paulo: Thomson, 2005 – Capítulo 5

Membros da Banca:

Ricardo Ferreira Martins	**Yuri Kaszubowski Lopes**
Carlos Norberto Vetorazzi Jr.	**Rui Jorge Tramontin Jr. (Presidente)**

UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

PROCESSO SELETIVO – 005 / 2022

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 3: Considere o modelo a seguir, que define a estrutura de um nó de uma Lista Simplesmente Encadeada para valores inteiros:

```
struct listNode{
      int value;
      struct listNode *next;
}
```

Implemente uma função em C que recebe como parâmetros o ponteiro para o primeiro nó da lista e o valor a ser inserido. A função deve fazer a inserção do valor ao final da lista.

```
struct listNode *insert( struct listNode *first, int info){
   struct listNode *newNode = malloc( sizeof(struct listNode) );
   newNode->value = info;
   newNode->next = NULL;
   // Se 'first' == NULL, lista está vazia --> retorna 'newNode' (único nó da lista). (20%)
   if( first = NULL ){
       return newNode;
   }
   else{         // Percorre até o último nó da lista...
      struct listNode *aux = first;
      while( aux->next != NULL ){
         aux = aux->next;
         // Insere 'newNode' no final da lista.
         aux->next = newNode;
         // Retorna o próprio 'first', pois continua sendo o primeiro nó da lista.
         return first;
      }
   }
}
```

TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. Estruturas de Dados Usando C. São Paulo: Makron Books, 1995. ISBN 85-346-0348-0

ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 3. ed. rev. e ampl. São Paulo: Pioneira Thomson Learning, 2011

Membros da Banca:

Ricardo Ferreira Martins	**Yuri Kaszubowski Lopes**
Carlos Norberto Vetorazzi Jr.	**Rui Jorge Tramontin Jr. (Presidente)**

PROCESSO SELETIVO – 005 / 2022

Área de Conhecimento: Ciência da Computação / Algoritmos e Estrutura de Dados

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 4: Considere uma Estrutura de Dados (ED) linear e ordenada pela ordem de inserção com as características conforme segue. Os dados são elementos homogêneos, e a ED é genérica, o tamanho dos elementos é informado no momento da criação da ED. Ao inserir o primeiro elemento **dado**, um nó, **no_t**, é alocado, este nó referencia uma região contígua da memória principal, **dados**, com capacidade para 4 elementos; dizemos que este nó tem capacidade para 4 elementos e a capacidade atual da ED é de 4 elementos. Quando o quinto elemento for adicionado, um novo nó com capacidade para 8 elemento é alocado, o quinto elemento é salvo neste novo nó; o novo nó tem capacidade para 8 elementos e a capacidade atual da ED passa a ser de 12 elementos. Sucessivamente, a ED cresce sempre que necessário durante a inserção, adicionando um novo nó com o dobro da capacidade do nó anteriormente adicionado.

Considere o descritor, **descritor_s** (ver Figura **4.a**), desta estrutura que armazena em **tamanho_dado** o tamanho dos elementos, valor recebido como argumento na função de criação **criar_array**. O descritor também possui uma referência, **cabeca** (ver Figura **4.a**), para o primeiro nó. Na estrutura do nó (**no_s**), **capacidade** indica a capacidade do nó (4, 8, 16, 32, 64, ...), **quantidade** indica o número de elementos do nó e **proximo** é um ponteiro para o próximo nó.

```c
typedef struct no_s {
    int capacidade;
    int quantidade;
    struct no_s *proximo;
    void *dados;
} no_t;

typedef struct descritor_s {
    int tamanho_dado;
    no_t *cabeca;
} descritor_t;
```

(4.a)

```c
void printdados(descritor_t *p) {
    no_t* no = p->cabeca;
    while (no != NULL) {
        for (int i = 0; i < no->quantidade; i++)
            printf("%i ", ((int*) no->dados)[i]);
        printf("\n");
        no = no->proximo;
    }
}

int main() {
    descritor_t *p = criar_array(sizeof(int));
    for (int i = 0; i < 28; i++)
        insere_array(p, &i);
    printdados(p);
    return 0;
}
```

(4.b)

Implemente as funções: **descritor_t* criar_array(int tamanho_dado)** que inicia (cria) esta ED, alocando e iniciando o descritor; e **int insere_array(descritor_t *p, void *dado)** que insere um novo item nesta ED, conforme descrito, e que insere novos nós conforme necessário. A implementação deve ser em Linguagem C. Você pode usar as funções **malloc**, **free** e **memcpy**. As demais funções auxiliares devem ser implementadas. O Código de teste acima (ver Figura **4.b**) deve gerar a saída abaixo:

```
0 1 2 3
4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
```

```c
// Alocação do descritor (20%)
descritor_t* criar_array(int tamanho_dado) {
    descritor_t *p = (descritor_t *) malloc(sizeof(descritor_t));
    if (p == NULL)
        return NULL;
    p->tamanho_dado = tamanho_dado;
    p->cabeca = NULL;
    return p;
}
// Restante do código... (80%)
no_t* aux_cria_no(int capacidade, int tamanho_dado) {
    no_t* no = (no_t *) malloc(sizeof(no_t));
    if (no == NULL)
        return NULL;  // ERRO
    no->capacidade = capacidade;
    no->quantidade = 0;
    no->proximo = NULL;
    no->dados = malloc(capacidade * tamanho_dado);
    if (no->dados == NULL) {
        free(no);
        return NULL;  // ERRO
    }

    return no;
}

int insere_array(descritor_t *p, void *dado) {
    if (p->cabeca == NULL) {
        p->cabeca = aux_cria_no(4, p->tamanho_dado);
        if (p->cabeca == NULL)
            return 1;
    }
    no_t* no = p->cabeca;
    while(no->capacidade == no->quantidade) {
        if (no->proximo == NULL) {
            no->proximo = aux_cria_no(2 * no->capacidade, p->tamanho_dado);
            if (no->proximo == NULL)
                return 1;
        }
        no = no->proximo;
    }

    int ajuste = p->tamanho_dado * no->quantidade;
    void *dest = (char*) no->dados + ajuste;
    memcpy(dest, dado, p->tamanho_dado);

    no->quantidade++;

    return 0; // 0 Erros
}
```

Membros da Banca:

Ricardo Ferreira Martins	**Yuri Kaszubowski Lopes**
Carlos Norberto Vetorazzi Jr.	**Rui Jorge Tramontin Jr. (Presidente)**

Assinaturas do documento

Código para verificação: **V48P2HI8**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:

RUI JORGE TRAMONTIN JUNIOR (CPF: 020.XXX.169-XX) em 10/07/2023 às 09:33:14
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:44:55 e válido até 30/03/2118 - 12:44:55.
(Assinatura do sistema)

YURI KASZUBOWSKI LOPES (CPF: 063.XXX.259-XX) em 10/07/2023 às 09:38:18
Emitido por: "SGP-e", emitido em 02/06/2021 - 13:17:43 e válido até 02/06/2121 - 13:17:43.
(Assinatura do sistema)

RICARDO FERREIRA MARTINS (CPF: 628.XXX.906-XX) em 10/07/2023 às 09:46:18
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:43:00 e válido até 30/03/2118 - 12:43:00.
(Assinatura do sistema)

CARLOS NORBERTO VETORAZZI JUNIOR (CPF: 070.XXX.978-XX) em 10/07/2023 às 09:51:32
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:34:55 e válido até 30/03/2118 - 12:34:55.
(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwMjgxMjJfMjgxNDVfMjAyM19WNDhQMkhJOA== ou o site https://portal.sgpe.sea.sc.gov.br/portal-externo e informe o processo **UDESC 00028122/2023** e o código **V48P2HI8** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.