Universidade Federal do ABC
Bacharelado em Ciência da Computação - 27/10/2017
Prova I de Programação Estruturada - Turma A1

NOME/RA : .....

Instruções

1. Escreva com caneta o seu nome completo e o RA;
2. As respostas devem ser transcritas com caneta esferográfica;
3. Cada questão vale 03 (três) pontos;
4. 01 ponto será dado caso siga os padrões descritos em sala de aula;
5. As questões serão corrigidas considerando corretude, padronização e estruturação do código.

Questões

Questão 1. Estruture o código abaixo de acordo com os conceitos aprendidos na disciplina [2,0 pts] e otimize-o [1,0 pt]:

```c
#include <stdio.h>

/* Persistencia multiplicativa de um numero eh quantas vezes
 * podemos multiplicar os digitos de um numero ate que ele
 * tenha apenas um digito.
 */
int  persistencia_multiplicativa(int n)
{
  unsigned long long  novo_n, quantas = 0;

  while (n>=10) {
    novo_n = 1;
    while (n != 0) {
      novo_n *= n%10;
      n = n/10;
    }
    quantas = quantas + 1;
    n = novo_n;
  }
  return quantas;
}


int main()
{
  printf("%d\n", persistencia_multiplicativa(679));
  return 0;
}
```

```c
#include <stdio.h>

int produto_digitos(int x)
{
  int prod = 1;

  /* colocar x!=0 requer uma opera  o de / e % extra */
  while (x>=10) {
    prod *= (x%10);
    x /= 10;
  }

```

```
  return x*prod;
}

/* Persistencia multiplicativa de um numero eh quantas vezes
 * podemos multiplicar os digitos de um numero ate que ele
 * tenha apenas um digito.
 */
int  persistencia_multiplicativa(int n)
{
  int quantas = 0;

  while (n>=10) {
    n = produto_digitos(n);
    quantas = quantas + 1;
  }
  return quantas;
}


int main()
{
  printf("%d\n", persistencia_multiplicativa(679));
  return 0;
}
```

Questão 2. Dado um número primo p, o número de Mersenne $2^p - 1$ é primo se somente se ele divide $S(p-1)$, com:

$$S(n)) = \begin{cases} 4, & \text{para n==1} \\ (S(n-1))^2 - 2, & \text{caso contrario.} \end{cases} \quad (1)$$

Implemente a função S em versão de recursão caudal [3,0 pts].

```
#include <stdio.h>

int S(int n)
{
  int sn1;

  if (n==1) return 4;

  sn1 = S(n-1);
  return sn1*sn1 - 2;
}

int STR(int n, int s)
{
  if (n==1) return s;

  return STR(n-1, s*s - 2);
}

int main ()
{
  printf("%d\n", S(5));
  printf("%d\n", STR(5, 4));

  return 0;
}
```

Questão 3. Corrija o código abaixo para que retorne o que é esperado [3,0 pts]:

```c
#include <stdio.h>

/* duplica s1 */
 char * strdup (char *s1)
 {

   int len = 0;
   char * dup;
   char * tmp;

   tmp = s1;  /* adiciona na correcao */
   while(*tmp++ != '\0') ++len;  /* troca s1 por tmp */

   dup = malloc(sizeof(char)*len);
   tmp = dup;

   if (dup != NULL) {
     while(*s1 != '\0') {
       *tmp = *s1;
       tmp++;
       s1++;
     }
   }

   return dup;
 }
int main(void)
{
  char s1[100] = "Ola\0jasdakjshdkjahsd";

  printf("s1 = %s\n", s1);
  printf("s1 duplicate = %s\n", strdup(s1));

  return 0;
}
```