Arquitectura de Computadores I

Ano Lectivo 2011/12 - 1º Semestre

Teste Prático II **13-Jan-2012**

Nº. Mec.:_____ **Nome:** _____

NOTE BEM: Leia atentamente todas as questões, comente o código usando a linguagem C e respeite a convenção de passagem de parâmetros e salvaguarda de registos que estudou. Na tradução para o *Assembly* do MIPS respeite rigorosamente os aspectos estruturais e a sequência de instruções indicadas no código original fornecido.
O código em C apresentado pode não estar funcionalmente correcto, pelo que **não deve ser interpretado**.

1) Codifique em *Assembly* do MIPS a seguinte função recursiva `eval()`:

```
double eval (int n)
{
  int i;
  double sum;
  if (n == 0)
     return 1.0;
  else
  {
     sum = 0.0;
     for (i = 0; i < n; i++)
        sum += eval (i);
     return (2.0 * sum / n) + n;
  }
}
```

Mapa de Registos

Label	Instrução em *Assembly* - Comentário em C	*Label*	Instrução em *Assembly* - Comentário em C

2) Preencha o quadro seguinte e codifique, em *Assembly* do MIPS, a função `main()` apresentada de seguida:

```
typedef struct
{
  char  name[18];
  float  price;
  char  flag;
  int  qtd;
} BOOK;

unsigned int exists (BOOK *, unsigned int, float, float *);
void read_array (BOOK *, unsigned int);
unsigned int atoi (char *);

int main (int argc, char *argv[])
{
  static BOOK book_array[10];
  static float max;

  unsigned int nelem, status;
  float value;

  if (argc < 1)
     return -1;

  nelem = atoi (argv[0]);
  value = read_float ();   /* SYSCALL */

  read_array (book_array, nelem);
  status = exists (book_array, nelem, value, &max);

  if (status == 1)
     print_string ("Pesquisa com sucesso");   /* SYSCALL */

  return 0;
}
```

***Offset* dos Campos**	
name	____
price	____
flag	____
qtd	____
***Size* of BOOK**	____

Mapa de Registos

Label	Instrução em *Assembly* - Comentário em C	*Label*	Instrução em *Assembly* - Comentário em C

3) Codifique em *Assembly* do MIPS a seguinte função `exists ()`:

```
unsigned int exists (BOOK *array, unsigned int nelem, float val, float *max)
{
  unsigned int i = 0;
  int index = -1;
  float mx = 0.0;
  while (i < nelem)
  {
     if ((array[i].price > mx) && (array[i].price <= val))
     {
        index= i;
        mx = array[i].price;
     }
     i++;
  }
  if (index != -1)
  {
     array[index].flag = 1;
     *max = mx;
     return 1;
  }
  else return 0;
}
```

Mapa de Registos

Label	Instrução em *Assembly* - Comentário em C	*Label*	Instrução em *Assembly* - Comentário em C

4) Codifique em *Assembly* do MIPS a seguinte função `total ()`:

```
float total (double *array, unsigned int nelem)
{
  unsigned int i = 1;
  float sum = 0.0;
  double *ptarray;

  for (ptarray = array; ptarray < (array + nelem); ptarray++)
  {
     if ((*ptarray > 0) || ((i % 2) != 0))
        sum += *ptarray * i;
     i++;
  }
  return (float) sum;
}
```

Mapa de Registos

Label	Instrução em *Assembly* - Comentário em C	*Label*	Instrução em *Assembly* - Comentário em C