Universidade Fernando Pessoa

Arquitectura de Computadores
Ficha de Exercícios nº3

Objectivos:

- Iniciação às funções e recursividade

1. Considere o seguinte programa em linguagem C e escreva o seu equivalente em assembly do MIPS[1].

```
int sumsquare(int i, int j)
{
      int sum;
      sum=((i*i)+(j*j));
      return sum;
}

main()
{
      int i=0, s=0;
      while(i<10){
            s+=sumsquare(i,i+1);
            i+=1;
      }
      printf("O valor final é %d\n",s);
}
```

2. Considere o seguinte programa em linguagem C. Escreva-o em assembly do MIPS.

```
int square(int i)
{
      return i*i;
}

int funct(int i,int j)
{
      int s;
      s=((i*square(i))-(j*square(j)));
      return s;
}

main()
{
      int i=0, s=0;
      while(i<5){
            s+=funct(5-i,i);
            i+=1;
      }
      printf("O valor final é %d\n",s);
}
```

3. Os programas seguintes apresentam uma versão iterativa e outra recursiva do cálculo do factorial de um número natural. Implemente-os em assembly do MIPS.

---

[1] Para efectuar o produto use a instrução `mul $r1, $r2, $r3`. Esta instrução executa o produto de `$r2` e `$r3` colocando o resultado em `$r1`.

```
int fact_iter(int n)
{
      int i, s=1;
      for(i=n;i>0;i--) s=s*i;
      return s;
}

int fact_recu(int n)
{
      switch(n){
            case 0:  return 1;
            default: return n*fact_recu(n-1);
      }
}


main()
{
      int n,facti,factr;

      printf("Indique um número inteiro positivo: ");
      scanf("%d",&n);

      facti=fact_iter(n);
      factr=fact_recu(n);

      printf("o factorial (iterativo) de %d é %d\n",n,facti);
      printf("o factorial (recursivo) de %d é %d\n",n,factr);

}
```

4. Considere a seguinte função recursiva em C. Que cálculo efectua? Implemente-a em assembly do MIPS.

```
int power(int val, unsigned pow)
{
        if (pow == 0)
                return 1;
        else
                return (power(val, pow - 1) * val);
}
```

Bibliografia:

[1] Patterson & Hennessy – *Computer Organization and Design: The hardware/software interface 4rd Ed* – MKP 2009.

University Fernando Pessoa

Computer Architecture
Exercise sheet nº3

Goals:

- Introduce the programming of functions and the use of recursion in MIPS assembly

1. Consider the following program written in C. Convert it to MIPS assembly [2].

```
int sumsquare(int i, int j)
{
      int sum;
      sum=((i*i)+(j*j));
      return sum;
}

main()
{
      int i=0, s=0;
      while(i<10){
            s+=sumsquare(i,i+1);
            i+=1;
      }
      printf("The final value is %d\n",s);
}
```

2. Consider the following functions written in C. Convert those functions to MIPS assembly.

```
int square(int i)
{
      return i*i;
}

int funct(int i,int j)
{
      int s;
      s=((i*square(i))-(j*square(j)));
      return s;
}

main()
{
      int i=0, s=0;
      while(i<5){
            s+=funct(5-i,i);
            i+=1;
      }
      printf("The final value is %d\n",s);
}
```

3. The following programs present an iterative solution and a recursive solution to calculate the factorial of a natural number. Implement both in MIPS assembly.

[2] To implement multiplication use the instruction `mul $r1, $r2, $r3`. This instruction executes the product between `$r2` and `$r3` putting the result in `$r1`.

```
int fact_iter(int n)
{
      int i, s=1;
      for(i=n;i>0;i--) s=s*i;
      return s;
}

int fact_recu(int n)
{
      switch(n){
            case 0:  return 1;
            default: return n*fact_recu(n-1);
      }
}


main()
{
      int n,facti,factr;

      printf("Introduce an integer positive: ");
      scanf("%d",&n);

      facti=fact_iter(n);
      factr=fact_recu(n);

      printf("Factorial (iterative) of %d is %d\n",n,facti);
      printf("Factorial (recursive) of %d is %d\n",n,factr);

}
```

4. Consider the following recursive function written in C. Which calculation does it perform? Convert the function to MIPS assembly.

```
int power(int val, unsigned pow) {
        if (pow == 0)
                return 1;
        else
                return (power(val, pow - 1) * val);
}
```

Bibliography:

[2] Patterson & Hennessy – *Computer Organization and Design: The hardware/software interface 4rd Ed* – MKP 2009.