

Programação em Assembly MIPS III

1. Escreva o código em assembly MIPS correspondente ao programa em C que se segue, preservando a sua estrutura funcional (versão recursiva ingénua dos números de Fibonacci).

```
int fib (int n) {
    if (n < 2)
        return n;
    else
        return fib(n - 1) + fib(n - 2);
}

int main() {
    int i;
    scanf("%d", &i);
    printf("fib(%d) = %d\n", i, fib(i));
    return 0;
}
```

2. Escreva o código em assembly MIPS correspondente ao programa em C que se segue, preservando a sua estrutura funcional (versão recursiva dos números de Fibonacci com memorização dos valores calculados).

```
int memFibs[1024] = {0,1,1};

int fib(int n) {
    if (memFibs[n] == 0)
        memFibs[n] = fib(n - 1) + fib(n - 2);
    return memFibs[n];
}

int main() {
    int i;
    scanf("%d", &i);
    printf("fib(%d) = %d\n", i, fib(i));
    return 0;
}
```

3. Escreva o código em assembly MIPS correspondente ao programa em C que se segue, preservando a sua estrutura funcional (cálculo do máximo divisor comum de forma recursiva).

```
int gcd(int i, int j) {
    if (j == 0)
        return i;
    else
        return gcd(j, i % j);
}

int main() {
    int i, j;
    scanf("%d", &i);
    scanf("%d", &j);
    printf("mdc(%d,%d) = %d\n", i, j, gcd(i, j));
    return 0;
}
```

4. (saber mais) Escreva o código em assembly MIPS correspondente ao programa em C que se segue, preservando a sua estrutura funcional (procura num vetor utilizando pesquisa binária).

```
int size = 15;
int v[] = {2, 9, 15, 26, 31, 37, 49, 51, 53, 54, 62, 66, 75, 84, 91};

int bsearch(int value, int low, int high) {
    int middle;
    if (low > high)
```

```
        return -1;
    middle = (low + high) / 2;
    if (value > v[middle])
        return bsearch(value, middle + 1, high);
    else if (value < v[middle])
        return bsearch(value, low, middle - 1);
    else
        return middle;
}

int main() {
    int i, index;
    scanf("%d", &i);
    index = bsearch(i, 0, size - 1);
    if (index == -1)
        printf("%d not found\n", i);
    else
        printf("%d found at index %d\n", i, index);
    return 0;
}
```