

113 計算機組織 Homework 2 1

Due date: 2024/10/21

- 1. 須符合測資。
- 2. MIPS code 和 C Code 結構相同:
  - (1) C Code 有 Loop，MIPS 須使用 jump/loop;
  - (2) C Code 有 Array，MIPS 須使用相對記憶體存取
  - (3) C Code 有 function，MIPS 須設計 function 並用 jal 呼叫
  - (4) function 參數變數,須使用 a 暫存器、一般區域變數,須使用 s 暫存器、回傳資料變數須使用 v 暫存器、迴圈索引暫時變數與儲存暫時運算結果使用 t 暫存器。
  - (5) 若為 caller function，使用 v, a, t 暫存器前須先將暫存器值存到 sp 要求的記憶體，之後要 restore。
  - (6) 若為 callee function，使用 s 暫存器前須將暫存器值存到 sp 要求的記憶體,之後要 restore。

請將每一題分別寫在一個 .asm 檔，EX: q1.asm  
將 4 個檔案壓縮成一個檔案，上傳到 I 學園

一、BMI Calculator (35%)

```
#include <stdlib.h>
#include <stdio.h>
int calculateBMI(int height, int weight) {
    int bmi = (weight * 10000) / (height * height);
    return bmi;
}
void printResult (int bmi) {
    if (bmi <= 17)
        printf("%s", "underweight\n");
    else if (bmi >= 25)
        printf("%s", "overweight\n");
    else
        printf("%d\n", bmi);
}
int main() {
    int height, weight, bmi;
    while (1) {
        scanf("%d", &height);
        if (height == -1)
            break;
        scanf("%d", &weight);
        bmi = calculateBMI(height, weight);
        printResult(bmi);
    }
    return 0;
}
```

| Input | Output      |
|-------|-------------|
| 170   | 18          |
| 53    | underweight |
| 180   | overweight  |
| 45    |             |
| 130   |             |
| 70    |             |
| -1    |             |
| 165   | 24          |
| 66    | underweight |
| 173   | 21          |
| 44    |             |
| 168   |             |
| 60    |             |
| -1    |             |
| 152   | overweight  |
| 69    | 24          |
| 199   | underweight |

|     |  |
|-----|--|
| 98  |  |
| 175 |  |
| 40  |  |
| -1  |  |

二、Recursion (35%)

```
#include <stdio.h>
void print(int size, int *x) {
    for (int i=0; i<size; i++) {
        printf("%d,",x[i]);
    }
}
int fact (int n, int *x) {
    int t=0;
    if (n < 2) {
        x[n] = t = 1;
        return t;
    }
    else {
        x[n] = t = fact(n - 1, x) + fact(n - 2, x);
        return t;
    }
}

int main() {
    int x[100];
    int n;
    x[0]=1;
    for (int i=1; i<100; i++)
        x[i]=0;
    scanf("%d", &n);
    fact(n, x);
    print(n, x);
    return 0;
}
```

| Input | Output             |
|-------|--------------------|
| 3     | 1,1,2,             |
| 5     | 1,1,2,3,5,         |
| 8     | 1,1,2,3,5,8,13,21, |

三、Selection sort (30%)

```
#include <stdlib.h>
#include <stdio.h>
void selectionSort(int array[], int n) {
    for (int i=0; i<n-1; i++) {
        int min_idx = i;
        for (int j=i+1; j<n; j++) {
            if (array[j] < array[min_idx]) {
                min_idx = j;
            }
        }
        int temp = array[min_idx];
        array[min_idx] = array[i];
        array[i] = temp;
    }
}
int main() {
    int array[5];
    for (int i = 0; i < 5; i++) {
        scanf("%d", &array[i]);
    }
    selectionSort(array, 5);
    for (int i = 0; i < 5; i++) {
        printf("%d\n", array[i]);
    }
}
```

```

return 0;
}

```

| Input | Output |
|-------|--------|
| 6     | 2      |
| 10    | 4      |
| 8     | 6      |
| 2     | 8      |
| 4     | 10     |
| 55    | 11     |
| 44    | 22     |
| 33    | 33     |
| 22    | 44     |
| 11    | 55     |
| 2     | 1      |
| 3     | 2      |
| 4     | 3      |
| 1     | 4      |
| 5     | 5      |

#### 四、Transpose of a Matrix (35%)

```

#include <stdlib.h>
#include <stdio.h>
void inputMatrix(int A[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            scanf("%d", &A[i][j]);
        }
    }
}
void transposeMatrixA1(int A[3][3], int T[3][3], int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            T[j][i] = A[i][j];
        }
    }
}
void transposeMatrixA2(int *B, int *T, int size) {
    int *ptrB, *ptrT, i;
    for (ptrB=B, ptrT=T, i = 1; ptrB<(B + (size*size)); ptrB++) {
        *ptrT = *ptrB;
        if (i < size) {
            ptrT += size;
            i++;
        }
        else {
            ptrT -= (size * (size - 1) - 1);
            i = 1;
        }
    }
}
void outputMatrix(int A[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }
}
int main() {
    int A[3][3];
    int transposeOfA1[3][3];
    int transposeOfA2[3][3];
    int *ptrA = &A[0][0];
    int *ptrTA2 = &transposeOfA2[0][0];
    inputMatrix(A);
    transposeMatrixA1(A, transposeOfA1, 3);
    transposeMatrixA2(ptrA, ptrTA2, 3);
    outputMatrix(transposeOfA1);
}

```

```

outputMatrix(transposeOfA2);
return 0;
}

```

| input | Output   |
|-------|----------|
| 1     | 1 4 7    |
| 2     | 2 5 8    |
| 3     | 3 6 9    |
| 4     | 1 4 7    |
| 5     | 2 5 8    |
| 6     | 3 6 9    |
| 7     |          |
| 8     |          |
| 9     |          |
| 11    | 11 22 33 |
| 44    | 44 55 66 |
| 77    | 77 88 99 |
| 22    | 11 22 33 |
| 55    | 44 55 66 |
| 88    | 77 88 99 |
| 33    |          |
| 66    |          |
| 99    |          |
| 63    | 63 43 79 |
| 72    | 72 88 51 |
| 58    | 58 56 12 |
| 43    | 63 43 79 |
| 88    | 72 88 51 |
| 56    | 58 56 12 |
| 79    |          |
| 51    |          |
| 12    |          |