Solution 1

1.5

a. P1: 3G/1.5CPI=2GIPS, P2: 2.5G/1CPI=2.5GIPS, P3: 4G/2.2CPI=1.818GIPS; P2 is the best

b. P1：3G\*10=10G, 2GIPS\*10s=20GI, P2: 2.5G\*10=25G, 2.5GIPS\*10s=25GI, P3: 4G\*10=40G, 1.818GIPS\*10s=18.18GI;

c. f=1.2/0.7=1.714, increasing the clock rate by 71.4%.

a.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | % | B | % | C | % | D | % | Average |
| P1 | 1 | 10 | 2 | 20 | 3 | 50 | 3 | 20 | 2.6 |
| P2 | 2 | 10 | 2 | 20 | 2 | 50 | 2 | 20 | 2 |

b. P1: 1e6\*2.6=2.6MI, P2: 1e6\*2=2MI

1.14

1.14.1 This is not possible

1.14.2 50\*1+110\*1+80\*4+16\*2=512, 80\*4-512/2=64, CPI=64/80=0.8

1.14.3 [512-(50\*1\*0.6+110\*1\*0.6+80\*4\*0.7+16\*2\*0.7)]/512=33.125%

c. double dz = 0 10000000010 010(1001) 41010(0)29 + 1 10000000011 1111101(1001)111 = 1 10000000011 0101000(1100)4(1001)71, z = 1 10000011 0101000(1100)31101 (=-21.100000381469727)

1.

a. An array arr of k integers

arr = (int \*) malloc(sizeof(int) \* k);

b. A string str of length p

str = (char \*) malloc(sizeof(char) \* (p + 1)); // Don't forget the null terminator!

c. An n x m matrix mat of integers initialized to zeros

mat = (int \*\*) calloc(n, sizeof(int \*));

for (int i = 0; i < m; i++) {

mat[i] = (int \*) calloc(m, sizeof(int));

}

6.

// Create a new stack with the given array size

struct stack \*init\_stack(int size) {

struct stack \*stk = (struct stack \*) malloc(sizeof(struct stack \*));

if (!stk) {

return NULL;

}

stk->size = size;

stk->topIndex = -1;

stk->stackArray = (int \*) malloc(sizeof(int) \* size);

if (!(stk->stackArray)) {

return NULL;

}

return stk;

}

// Add the given element to the stack. Resize by doubling the array size if full.

void push(int x, struct stack\* stk) {

if (stk->topIndex == stk->size - 1) {

int \*stackArray = (int \*) realloc(stk->stackArray, stk->size \* 2)

if (!stackArray) {

return;

}

stk->stackArray = stackArray;

stk->size = stk->size \* 2;

}

stk->topIndex += 1;

stk->stackArray[stk->topIndex] = x;

}

// Remove the top element from the stack. Return 0 if empty.

int pop(struct stack \*stk) {

if (stk->topIndex < 0) {

return 0;

}

int x = stk->stackArray[stk->topIndex];

stk->stackArray[stk->topIndex] = 0;

stk->topIndex -= 1;

return x;

}

7. 