**★★★☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：10033 Interpreter**

**解題者：何彥霆**

**解題日期：2023年4月6日**

**題意：**

 按照題目所給的規則設計一個Interpreter，假設此計算機有10個register和1000 words大小的RAM，所有instruction跟address都以一個3-digit的數字表示，以下是相關規則：

• 100 means halt

• 2dn means set register d to n (between 0 and 9)

• 3dn means add n to register d

• 4dn means multiply register d by n

• 5ds means set register d to the value of register s

• 6ds means add the value of register s to register d

• 7ds means multiply register d by the value of register s

• 8da means set register d to the value in RAM whose address is in register a

• 9sa means set the value in RAM that address is in register a to the value of register s

• 0ds means goto the location in register d unless register s contains 0

 最終輸出為一數字，代表總計執行了多少個instructions，包括halt。

**題意範例：**

Input: 299 492 495 399 492 495 399 283 279 689 078 100 000 000 000

Count PC Instruction

 1 1 299 $9 -> 9

 2 2 492 $9 -> 18

 3 3 495 $9 -> 90

 4 4 399 $9 -> 99

 5 5 492 $9 -> 198

 6 6 495 $9 -> 990

 7 7 399 $9 -> 999

 8 8 283 $8 -> 3

 9 9 279 $7 -> 9

10 10 689 $8 -> 2

11 11 078 PC goto 9

12 10 689 $8 -> 1

13 11 078 PC goto 9

14 10 689 $8 -> 0

15 11 078 PC goto 9

16 12 100 halt

Output: 16

**解法：**

建reg[10], ram[1000], 把opcode, d, n, s從各個input instruction提出來，並針對不同的opcode進行分類操作即可。

switch (opcode) {

case 0:

 if (regs[s] != 0) {

 pc = regs[d];

 }

 break;

case 2:

 regs[d] = n;

 break;

case 3:

 regs[d] = (regs[d] + n) % 1000;

 break;

case 4:

 regs[d] = (regs[d] \* n) % 1000;

 break;

case 5:

 regs[d] = regs[s];

 break;

case 6:

 regs[d] = (regs[d] + regs[s]) % 1000;

 break;

case 7:

 regs[d] = (regs[d] \* regs[s]) % 1000;

 break;

case 8:

 regs[d] = ram[regs[s]];

 break;

case 9:

 ram[regs[s]] = regs[d];

 break;

case 1:

 cout << instructions << endl;

 out\_flag = 1;

}

**解法範例：**

無

**討論：**

這題沒什麼算法難點，比較看重的是計算機組織的基礎知識，看懂後按功能實現即可。

**程式：**

無