Lab 6a Report

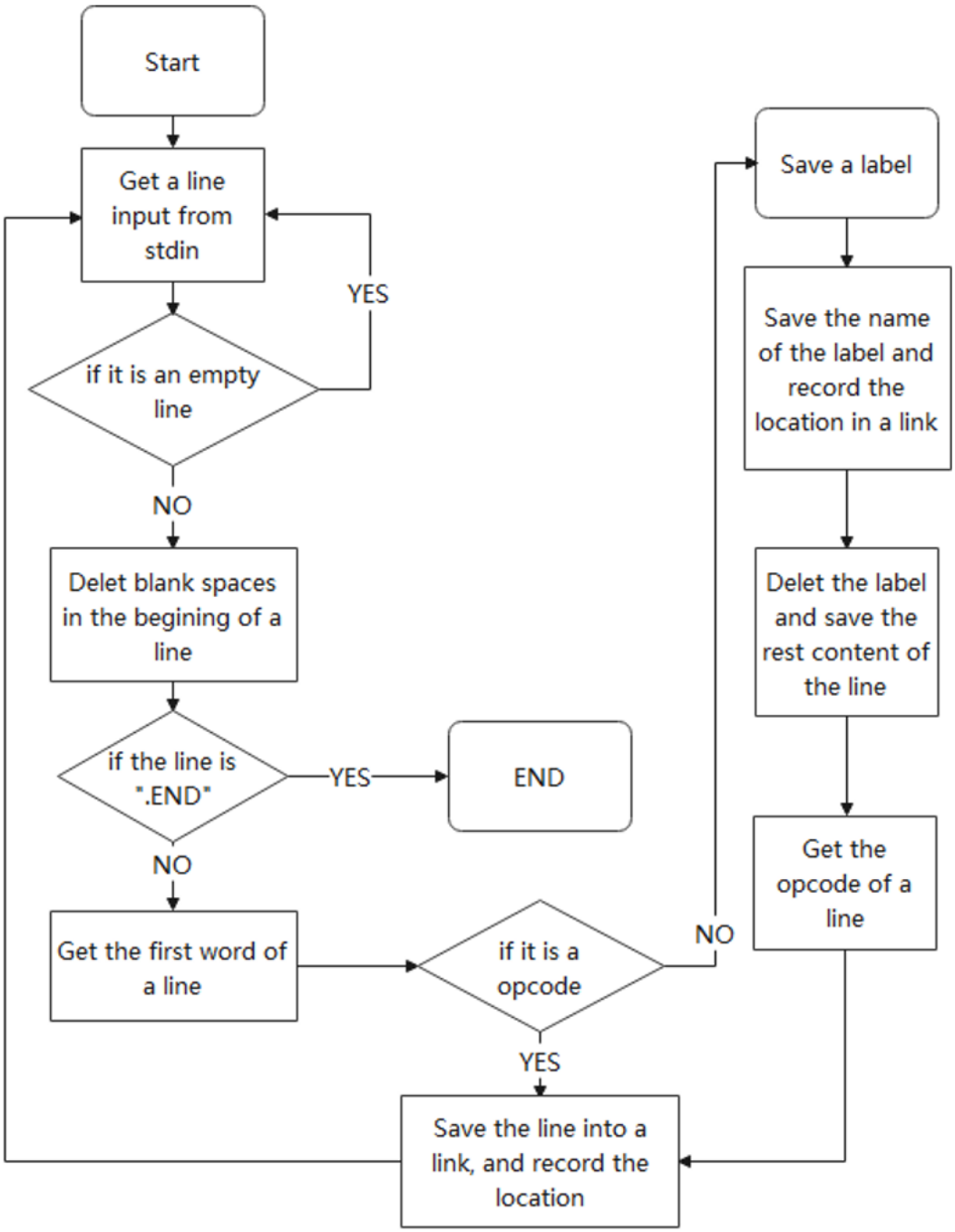
Student name: 王晨雨

Strudent ID: 3200102324

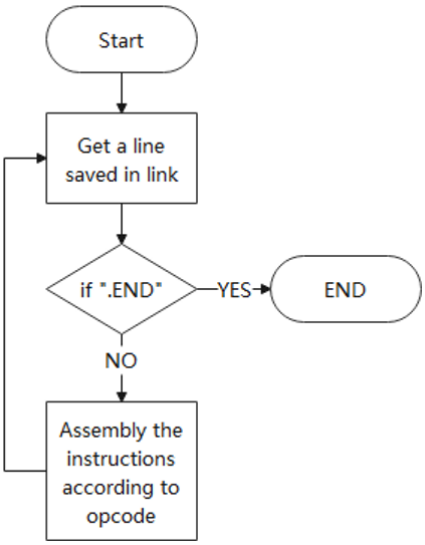
Report submission date: 2021/8/18

1. Algorithm

- Process the input from *stdin*

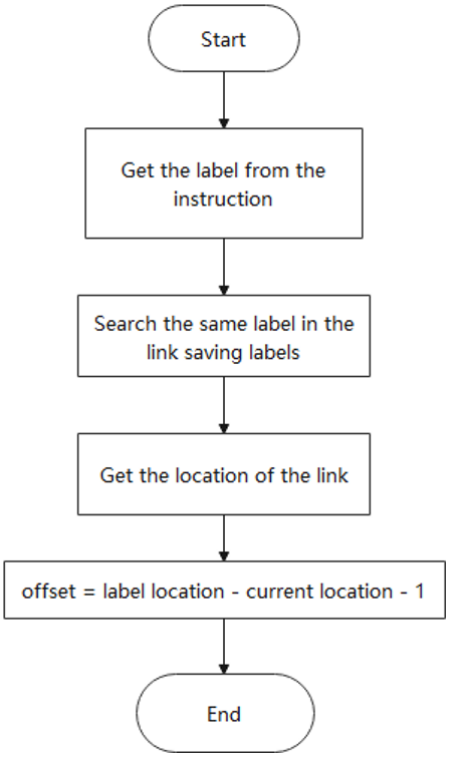


- The record of the location of every line of instruction:
  - Usually, when we process the next line, `location++;`
  - If we meet `.BLKW #n` , then when we process the next line, `location += n;`
  - If we meet `.STRINGZ "str[n]"` , then when we prosee the next line, `location+=strlen(str)+1;`
- The process of assembly



(The detailed code of Assembly can be found in PTA)

- The output of the binary code:
  - I use `bitset` to specified output format.
- The use of language: C++
- Calculate the offset of label



2. Essential parts of code

Some essential structure:

```
/* struct label */
/* Usage: save a label and record the location of it */
struct label{
    char name[20];
    int location;
    struct label *next;
};

/* struct linecontent */
/* Usage: save the input content of lines */
struct linecontent{
    char linechar[100];
    int no; /* The location of lines */
    struct linecontent *next;
};
```

Some import functions:

```
/*
 * Funtion name: DecimaltoBin
 * Translate a string of numbers into a decimal number
 * Usage: int DecimaltoBin(char *decimal)
 * Return the result of translation
 */
int DecimaltoBin(char *decimal)
{
    int length = strlen(decimal);
    int i, k;
    k = 0;
    int result = 0;
    if(decimal[0] != '-')
    {
        for(i = length; i > 0; i--)
        {
            result = result + pow(10, i - 1) * (decimal[k] - 48);
            k++;
        }
    }
    else
    {
        length--;
        for(i = length; i > 0; i--)
        {
            result = result + pow(10, i - 1) * (decimal[k+1] - 48);
            k++;
        }
        result = - result;
    }
    return result;
}

/*
 * Funtion name: Hex
```

```
 * Translate a string of numbers into a hexadecimal number
 * Usage: int Hex(char *hex)
 * Return the result of translation
 */
int Hex(char *hex)
{
    int result = 0;
    int length = strlen(hex);
    int i, k;
    k = 0;
    if(hex[0] == '-') /* it is a negative hex imm */
    {
        length--;
        for(i = length; i > 0; i--)
        {
            if('0' <= hex[k+1] && hex[k+1] <= '9')
                result = result + pow(16, i - 1) * (hex[k+1] - 48);
            else
                result = result + pow(16, i - 1) * (hex[k+1] - 'A' + 10);
            k++;
        }
        result = -result;
    }
    else
    {
        for(i = length; i > 0; i--)
        {
            if('0' <= hex[k] && hex[k] <= '9')
                result = result + pow(16, i - 1) * (hex[k] - 48);
            else
                result = result + pow(16, i - 1) * (hex[k] - 'A' + 10);
            k++;
        }
    }
    return result;
}
```