

This Week

Adding more structure and data types to your programs.

more loops,  
if/else structures,  
complex programs,  
flowcharts,  
subroutines + stack

bytes, halfwords,  
ascii (text), etc.

Homework:

music player - halfword data types  
- more complex structure  
to program  
- lots of data

Final Project:

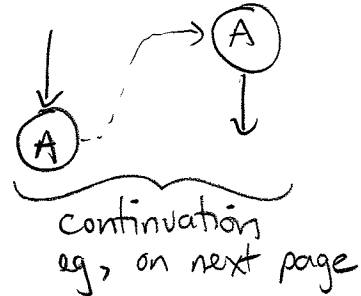
eg, real working alarm clock  
- keeps time  
- set alarm/time  
- play music on alarm  
- snooze

Flowcharts help plan + document algorithm

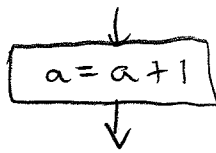
get algorithm correct first  
get assembly language correct second

sequence of steps

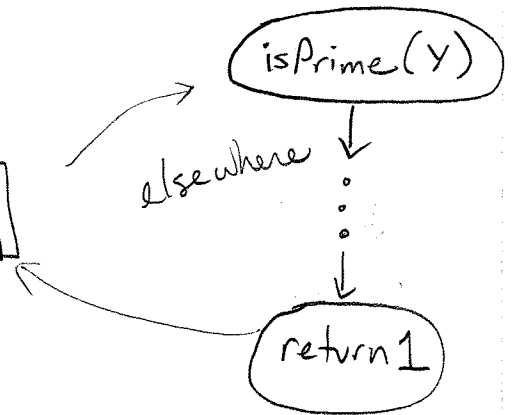
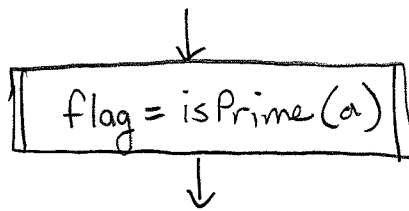
terminals



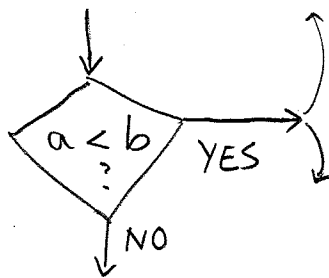
simple step



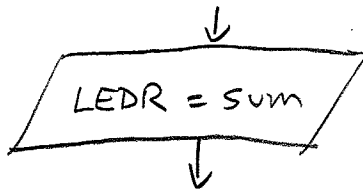
complex step  
= subroutine



decision



input/output



- Main Ideas
- ① flow chart helps us detect algorithm bugs
  - ② easy to translate each flowchart box into 2-3 assembly instructions
  - ③ verify ① and verify ② ⇒ correct program

Example

compute  $\sum_{i=1}^N i$  and display sum on LEDR

• global -start

• text

```
-start:  movia  r16, sum
        movi  r17, 0
        stw   r17, 0(r16)
```

```
        movia  r18, N
        ldw   r19, 0(r18)
```

```
loop:   add   r17, r17, r19
```

```
        addi  r19, r19, -1
```

```
bgt.   r19, r0, loop
```

```
        movia  r23, IOBASE
        stwio  r17, LEDR(r23)
        stw   r17, 0(r16)
```

```
STOP:   br    STOP
```

• data

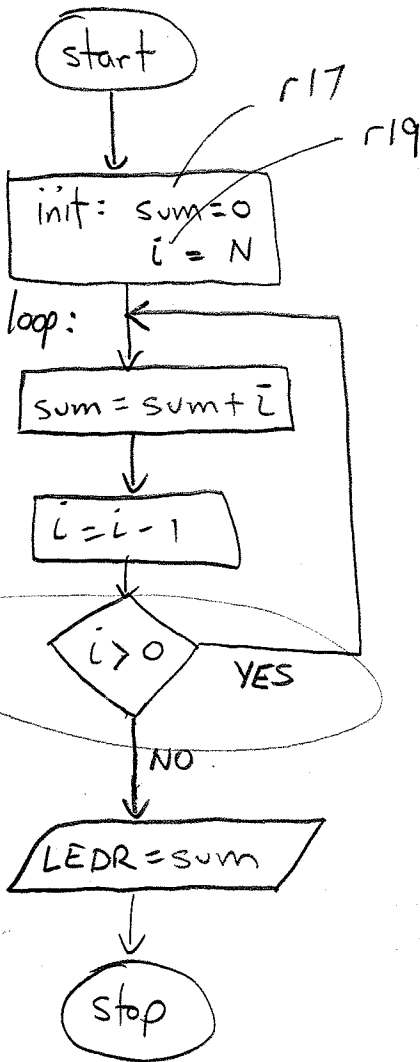
N:

• word 20

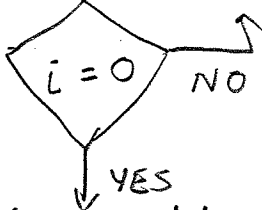
sum:

• skip 4

• end



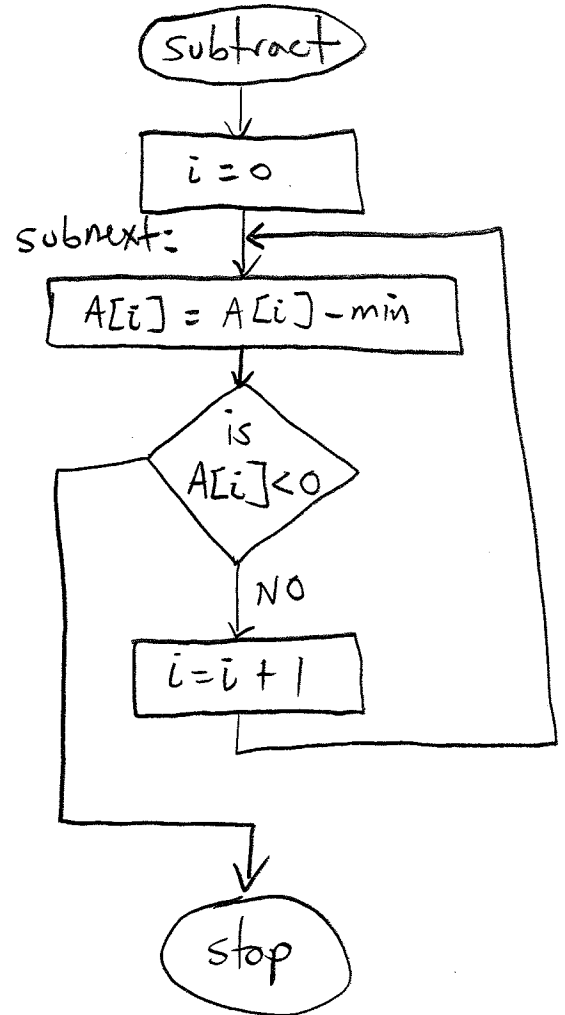
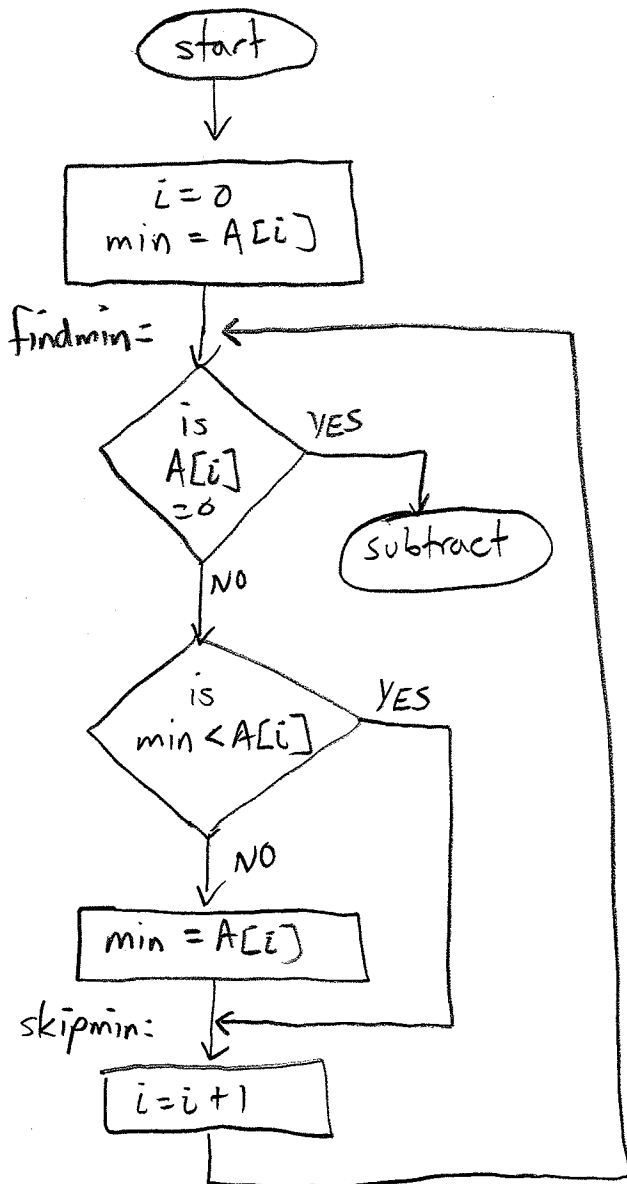
Why is the following not as good?



(Hint: what if N=0...)

Note: this algorithm is still not as "good" or perfect as it could be: ① if N is signed, it does not work for N < 0 ② if N is unsigned, then i must also be unsigned and "bgt" must change to "bgtu", but a bug exists at N=0 => how to fix?

Example given an array where the last element (end of array) is  $\emptyset$ , subtract the smallest element from all elements



NOTE: this subtraction loop has a minor / subtle logic error in the algorithm. What is the error? (X)

You probably wouldn't catch the error by looking at the assembly code itself.

(X) When  $A[0] = 0$  (an empty array), the loop goes past the end of the array and may waste CPU time, or worse like trigger I/O that shouldn't happen.