Problem 1

* Problems with the existing flowchart:
  1. If \( N = 1 \), then (on first pass, when \( i = 1 \)),
     the command "DIFF = \( A_{i+1} - A_i \)" requires
     the value \( A_2 \), which doesn't exist.
  2. When \( i = N - 1 \), the decision step "IS \( i > N \) ?"
     returns "NO"; then \( i \) is incremented to \( N \);
     then the command "DIFF = \( A_{i+1} - A_i \)" calls
     the value \( A_{N+1} \), which doesn't exist.

* New flowchart:

START

INPUT \( A_i \) values

INPUT \( N \)

IS \( N = 1 \) ?

YES

\( i = 1 \) →

DIFF = \( A_{i+1} - A_i \)

NO

\( i = i + 1 \)

IS DIFF > 0?

YES

\( i = i + 1 \)

NO

OUTPUT 'YES'

IS \( i < N - 1 \) ?

YES

END

NO

\( i = i + 1 \)

OUTPUT 'END'

END

STOPs loop \( @ i = N - 1 \)
Loop

\[ i = i + 1 \]

Is \( i \geq n \)?

\[ \text{SUM} = \text{SUM} + C_i \]

Is \( \text{SUM} \geq 100 \)?

If \( \text{SUM} \geq 100 \), output "sufficient"

Change = SUM - 100

IF Change > 0

Output "one change is"

\[ \text{OUTPUT } \text{Change } \text{CENTS} \]

Loop

\[ i = i + 1 \]

End

Problem 2
Flowchart

Input Values
Read \( C_i \)
Start

\[ i = 1 \]

\[ \text{SUM} = 0 \]

Repeat steps of value of coins inserted

Output "The first "

Change < 0

Output "the first change is"

\[ \text{OUTPUT } \text{Change } \text{CENTS} \]
Problem 2b

Two sample codes. The first uses a for-loop as we saw in class:

```matlab
% Assume the vector C is already in the workspace
N = length(C);
% 'sum' keeps track of the value of the coins inserted in the machine; 'number' is zero initially, then gives the minimum number of coins required; 'amount' gives the change returned when the minimum number of coins is inserted
sum = 0;
number = 0;
% Loop through C vector
for i = 1:N
    sum = sum + C(i);
    if sum >= 100 & number == 0
        % Run these statements as soon as sum gets to 100
        number = 1;
        amount = sum-100;
    end
end
if sum < 100
    fprintf('Sorry, insufficient funds.
')
else
    fprintf('The first %d coins are sufficient.
',number)
    fprintf('Your change is %d cents.
',amount)
end
```

The second uses the 'break' command (which we didn't cover in class), which kicks you out of the for-loop at a given time:

```matlab
% Assume the vector C is already in the workspace
N = length(C);
% 'sum' keeps track of the value of the coins inserted in the machine; 'number' gives the minimum number of coins required; 'amount' gives the change returned when the minimum number of coins is inserted
sum = 0;
% Loop through C vector
for i = 1:N
    sum = sum + C(i);
    if sum >= 100
        % Run these statements as soon as sum gets to 100
        number = 1;
        amount = sum-100;
        break;
    end
end
if sum < 100
    fprintf('Sorry, insufficient funds.
')
else
    fprintf('The first %d coins are sufficient.
',number)
    fprintf('Your change is %d cents.
',amount)
end
```

Problem 2c

Two sample codes. The first uses a while-loop on the index `i` that runs through the whole vector `C`:

```matlab
% Assume the vector C is already in the workspace
N = length(C);
% 'sum' keeps track of the value of the coins inserted in the machine; 'number' gives the minimum number of coins required; 'amount' gives the change returned when the minimum number of coins is inserted
sum = 0;
% While-loop through C vector. Variable 'i' keeps track of number of coins inserted
i = 1;
while i <= N
    sum = sum + C(i);
    if sum >= 100
        % Run these statements as soon as sum gets to 100
        number = 1;
        amount = sum-100;
        break;
    end
    i = i+1;
end
```
if sum < 100
    fprintf('Sorry, insufficient funds.
')
else
    fprintf('The first %g coins are sufficient.
', number)
    fprintf('Your change is %g cents.
', amount)
end

The second code uses a while-loop conditioned on both the index i and the value of the coins inserted:

% Assume the vector C is already in the workspace

N = length(C);

% 'sum' keeps track of the value of the coins inserted
% in the machine; 'number' gives the minimum number of
% coins required; 'amount' gives the change returned
% when the minimum number of coins is inserted

sum = 0;

% While-loop through C vector. Variable 'i' keeps track of
% number of coins inserted

i = 1;

while i <= N & sum < 100
    % These statements run as long as sum is under 100
    sum = sum + C(i);
    number = i;
    amount = sum - 100;
    i = i + 1;
end

if sum < 100
    fprintf('Sorry, insufficient funds.
')
else
    fprintf('The first %g coins are sufficient.
', number)
    fprintf('Your change is %g cents.
', amount)
end