


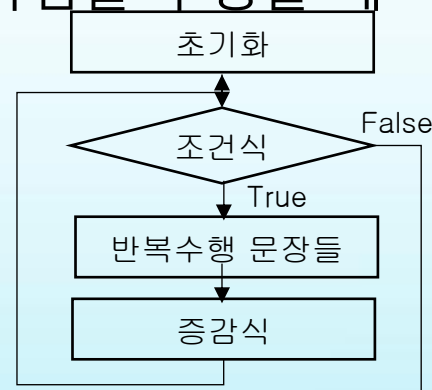
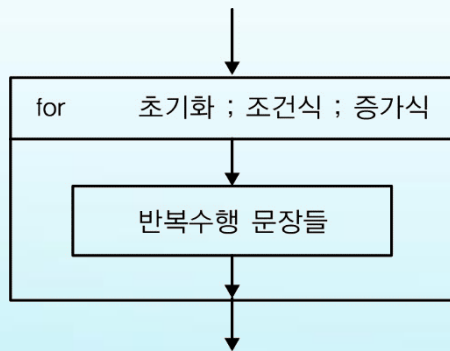


4장 반복

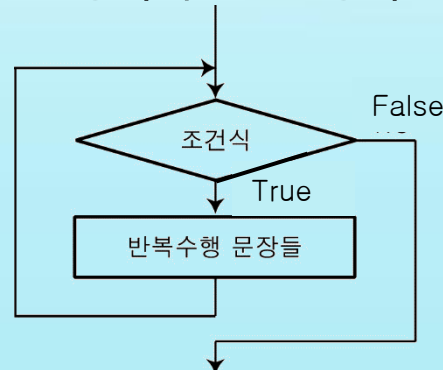
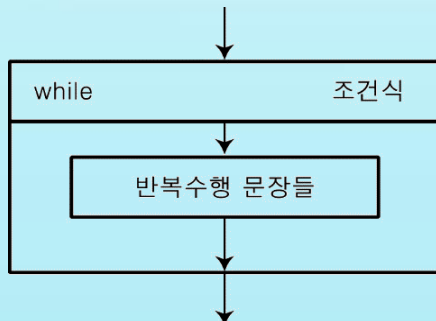
- 4.1 일정한 횟수 반복(for)
 - 4.2 선조건 반복(while)
 - 4.3 후조건 반복(do-while)
 - 4.4 중첩 반복
- 

4.1 반복 기초

- 일반적으로 특정 문장이나 여러 문장의 블록을 여러 번 반복 수행하고 싶을 경우 반복 구조를 사용한다.
- 반복 구조의 유형으로는 크게 두가지로 나누어 진다.
 - 반복 횟수가 정해져 있는 경우
 - 조건에 따라 반복 수행해야 하는 경우
- 횟수가 정해져 있는 반복적인 작업을 수행할 때 사용

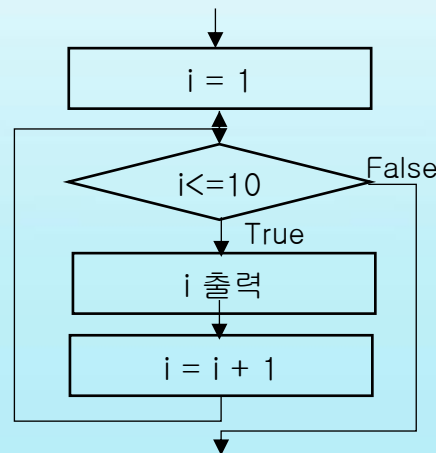
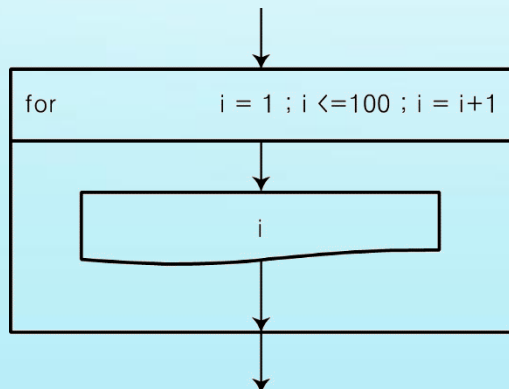


- 판단 조건이 따라 반복 수행하는 경우(참일 경우 반복, 거짓이면 벗어남)



4.1 일정한 횟수 반복 기초

- 횟수가 정해져 있는 반복적인 작업을 수행할 때 사용
- 1부터 10까지 반복하면서, 출력하는 순서를 예로 들면 다음과 같다
 - ① 변수 i 를 선언하고 1로 초기화($i = 1$)
 - ② 조건식 수행, 변수 i 가 10보다 작거나 같은지 비교($i \leq 10$)
 - ③ 조건식이 참이면 다음 문장, 거짓이면 반복 수행 종료
 - ④ 반복 문장 수행, i 를 출력
 - ⑤ 증감 문장 수행, i 를 1 증가($i = i + 1$)
 - ⑥ ②부터 다시 수행

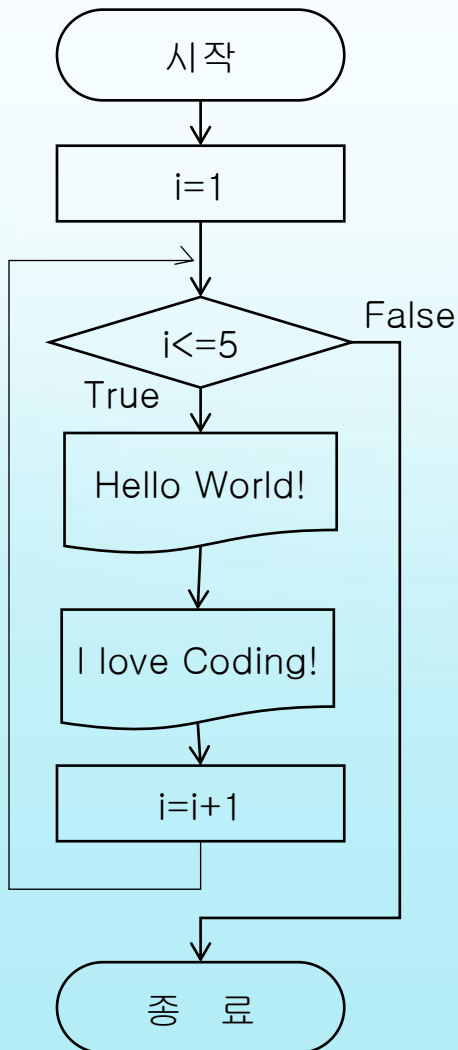


```
int i;  
for(i=1; i<=10; i=i+1)  
    printf("%d ", i);
```

4.1 일정한 횟수 반복 기초_예제 1

"Hello World!"를 5번 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



pseudocode

```
1  main() {  
2      int i;  
3      for( i = 1; i <= 5; i = i + 1 ) {  
4          printf("Hello World! ");  
5          printf("I love Coding!\n");  
6      }  
7  }
```

실행 예

```
Hello World! I love Coding!  
Hello World! I love Coding!  
Hello World! I love Coding!  
Hello World! I love Coding!  
Hello World! I love Coding!
```

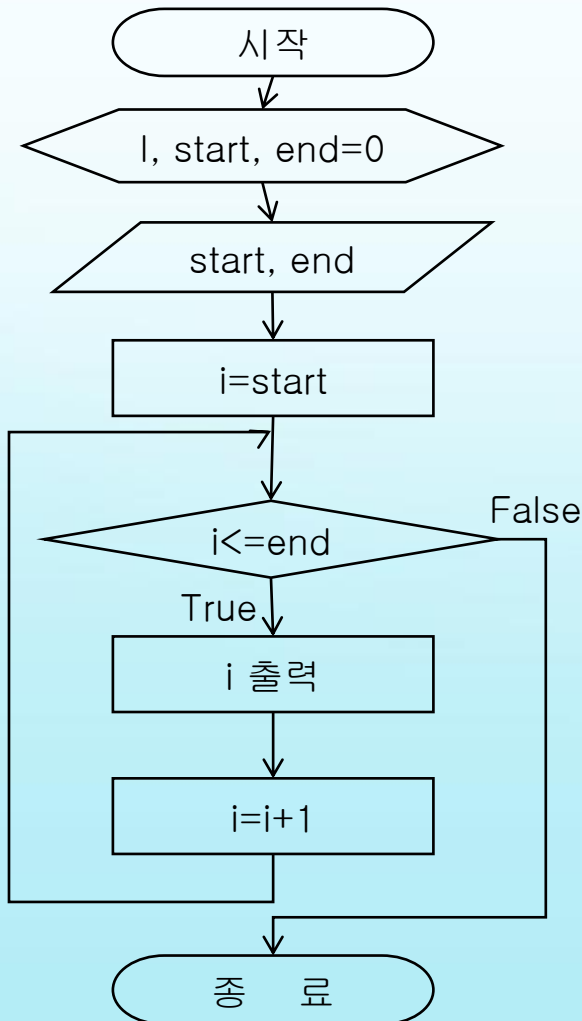
생각해 보기

- Hello World 앞에, 반복 횟수도 함께 출력하도록 수정해보자.
1. Hello World! I love Coding!
- 반복 횟수를 입력 받아 수행하도록 수정해보자.

4.1 일정한 횟수 반복 기초 _예제 2

두 수를 입력받아 두 수 사이의 값들을 출력하는 순서도와 슈도 코드를 작성하시오.

flowchart



pseudocode

```
1  main() {  
2      int i, start, end;  
3      scanf("%d %d", &start, &end);  
3      for( i = start; i <= end; i = i + 1 ) {  
4          printf("%d ", i);  
5      }  
6  }
```

실행 결과

```
1 10  
1 2 3 4 5 6 7 8 9 10
```

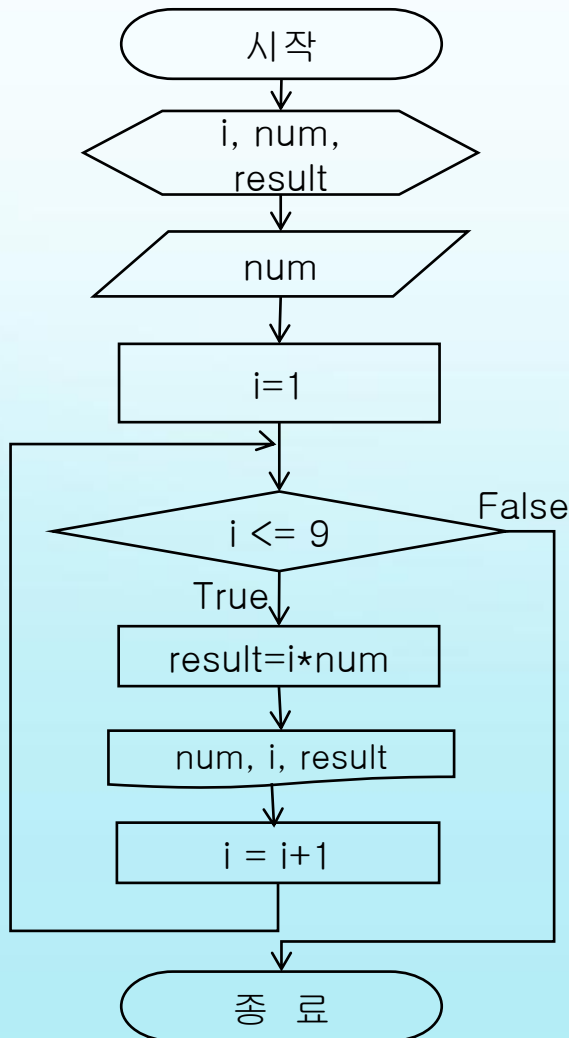
생각해 보기

두 수 사이에 홀수인 경우만 출력하도록 수정해보자.

4.1 일정한 횟수 반복 기초 _예제 3

입력된 수의 구구단을 출력하는 순서도와 슈도코드를 작성하십시오.

flowchart



pseudocode

```
1  main() {  
2      int i, num, result;  
3      scanf("%d", &num);  
3      for( i = 1; i <= 9; i = i + 1 ) {  
4          result = i * num;  
5          printf("%d * %d = %d\n ", num, i, result);  
6      }  
7  }
```

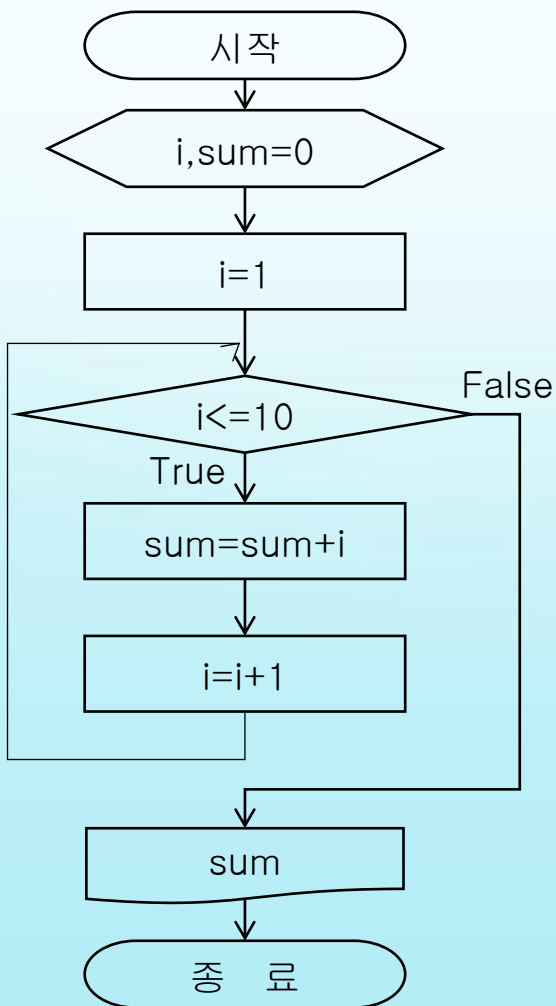
실행 결과

```
3  
3 * 1 = 3  
3 * 2 = 6  
3 * 3 = 9  
3 * 4 = 12  
3 * 5 = 15  
3 * 6 = 18  
3 * 7 = 21  
3 * 8 = 24  
3 * 9 = 27
```

4.1 일정한 횟수 반복 기초 _예제 4

1부터 10까지 합을 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



pseudocode

```
1  main() {  
2      int i, sum=0;  
3      for( i = 1; i <= 10; i = i + 1 ) {  
4          sum = sum + i;  
5      }  
6      printf("합: %d \n", sum);  
7  }
```

실행 결과

합: 55

생각해 보기

- 누적 변수인 sum을 0으로 초기화한 이유는 무엇인가?
- 두 수를 입력 받아, 두 수 사이에 합을 출력하도록 수정해보자.

실행과정

i	sum
	0
1	1
2	3
3	6
4	10
5	15
6	21
7	28
8	36
9	45
10	55
11	

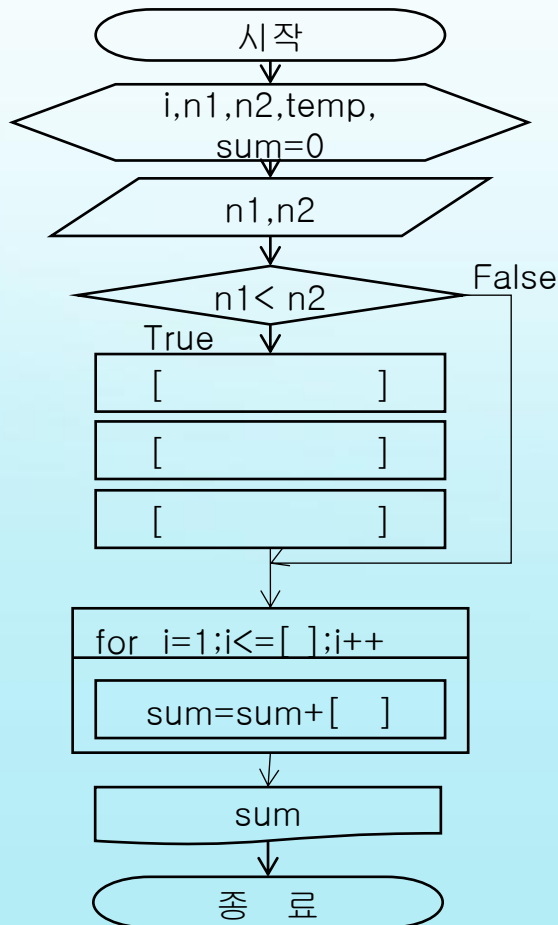
4.1 일정한 횟수 반복 기초 _문제 1

곱셈은 덧셈을 편리하게 하기 위해 등장한 연산이다. 덧셈으로 곱셈을 하는 슈도코드를 작성하시오.

힌트

3 * 9 일 경우, 3을 9번 더하는 것과 9를 3번 더하는 것 중 어느 것이 효율적인가?

flowchart



pseudocode

```
1  main() {
2      int i, n1, n2, temp, sum=0;
3      scanf("%d %d", &n1, &n2);
4      if (n1 < n2) {
5          [ ]
6          [ ]
7          [ ]
8      }
9      for( i = 1; i <= [ ]; i = i + 1 ) {
10         sum = sum + [ ];
11     }
12     printf("곱: %d \n", sum);
13 }
```

실행 결과

10 3
곱: 30

4.1 일정한 횟수 반복 연습

- for문을 이용해 1부터 100까지 누적 합을 구하는 것을 앞에서 학습했다. 누적은 수학에서 시그마(Σ)로 표현하며, 다음과 같다.
 - $a_1 + a_2 + a_3 + \dots + a_n = \sum_{k=1}^n a_k$
- 이것은 등차수열로, 이러한 등차수열의 성질을 이해하면, 반복 횟수를 줄일 수 있는 알고리즘을 사용할 수 있다.
 - (1) 1, 2, 3,...+100 은 첫 번째 항이 1, 공차가 1이다.
k번째 항은 $1 + (k - 1) * 1 \rightarrow 1 + k - 1 \rightarrow k$
 - (2) $k = 100$
 - (3) 따라서, $1+2+3+\dots+100$ 은 $= \sum_{k=1}^{100} k$
- 등차 수열의 성질을 이용해보자
- 예를들어, $2 + 5 + 8 + \dots + 41$ 을 구한다고 할 때 단순히 for를 이용하면 다음과 같이 알고리즘을 기술할 수 있다.

```
1 main() {  
2     int i, sum=0;  
3     for(i=2; i<=41; i=i+3)  
4         sum = sum + i;  
5 }
```

4.1 일정한 횟수 반복 연습

- $2 + 5 + 8 + \dots + 41$ 등차 수열로 표현해 알고리즘을 기술해보자.
 - (1) $2 + 5 + 8 + \dots + 41$ 은 첫째항이 2, 공차가 3인 등차 수열이다.
k번째 항은 $2 + (k - 1) * 3 \rightarrow 2 + 3k - 3 \rightarrow 3k - 1$
 - (2) $3k - 1 = 41 \rightarrow 3k = 42 \rightarrow k = 14$
 - (3) 따라서, $2 + 5 + 8 + \dots + 41$ 은 $= \Sigma(3k-1)$
- 시그마식을 알고리즘으로 기술하면 다음과 같으며, 두 가지 알고리즘의 차이점에 대해 생각해보자.
- 두 가지 알고리즘의 반복 횟수는 동일하며, 상황에 따라 사용할 수 있는 다양한 구조를 익히는 것이 중요하다,

반복문의 특징	수학적 모델
<pre>1 main() { 2 int i, sum=0; 3 for(i=2; i<=41; i=i+3) 4 sum = sum + i; 5 }</pre>	<pre>1 main() { 2 int i, sum = 0; 3 for(i=1; i<=14; i++) 4 sum = sum + (3 * i - 1); 5 }</pre>

4.1 일정한 횟수 반복 연습_문제 1

1부터 10까지의 수 중에서 3의 배수의 합을 구하는 순서도와 슈도 코드를 완성하여라.

참고

1. 시그마() 식으로 나타내고, 이 식을 이용한 슈도코드를 작성해보자
2. 3의 배수이므로 3,6,9,...

pseudocode

```
1  main() {  
2      int i, sum = 0;  
3      for(i = 3; i <= 10; i = i + 3)  
4      {  
5          [                ];  
6      }  
7      printf("3의 배수의 합은 %d이다.", sum);  
8  }
```

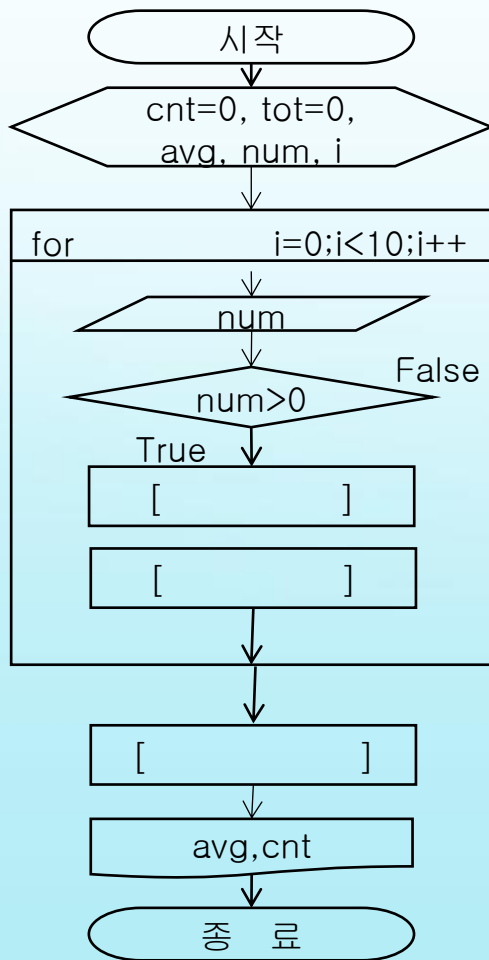
실행 예

```
3 6 9  
3의 배수의 합은 18이다.
```

4.1 일정한 횟수 반복 연습_문제 2

10개의 숫자를 입력 받아 0보다 큰 숫자에 대한 평균과 개수를 출력하는 순서도와 슈도코드를 완성하시오.

flowchart



pseudocode

```
1  main() {
2      int cnt = 0, tot = 0, ave, num, i;
3      for(i = 0; i < 10; i++) {
4          scanf("%d", &num);
5          if( num > 0) {
6              [ ]
7              [ ]
8          }
9      }
10     [ ]
11     printf("%d%d", ave, cnt);
12 }
```

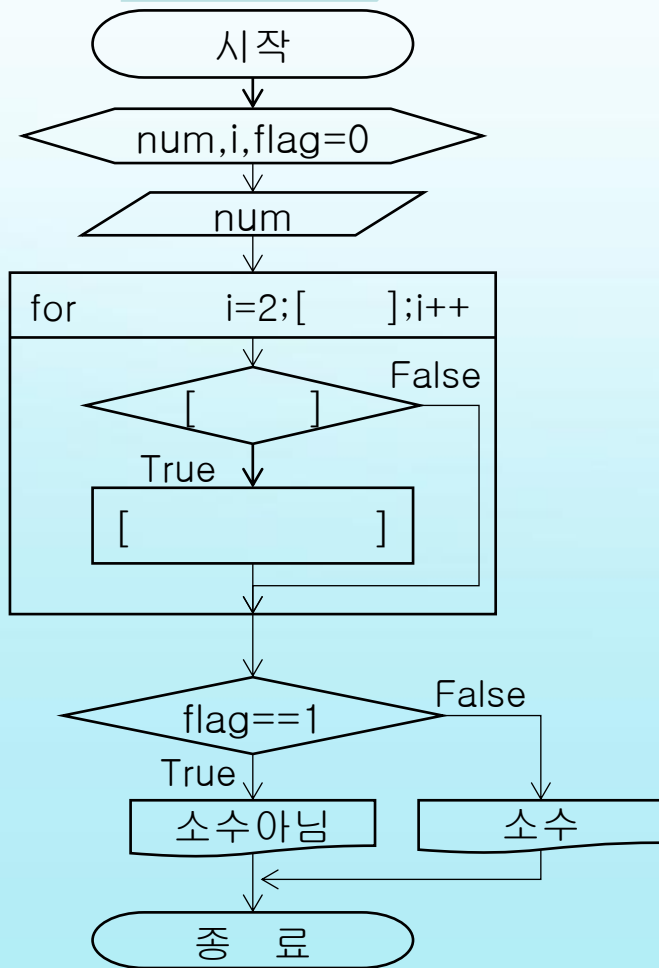
4.1 일정한 횟수 반복 연습 _문제 3

입력된 값이 소수인지 아닌지 판별하는 알고리즘의 순서도와 슈도 코드를 완성하시오.

참고

소수는 1과 자신 이외에는 더 이상 나누어 떨어지지 않는 수로 1은 제외한다.

flowchart



pseudocode

```
1  main(){
2      int num, i, flag = 0;
3      scanf("%d", &num);
4      for(i = 2; [ ] ; i++){
5          if([ ] ) {
6              [ ] ;
7              break;
8          }
9      }
10     if(flag == 1)
11         printf("소수 아님");
12     else
13         printf("소수 ");
14 }
```

생각해 보기

break문이 있을 경우와 없을 경우의 차이점을 생각해 보자.



4.1 일정한 횟수 반복 연습 _문제 4

10개의 데이터를 입력 받아 최소값(min), 최대값(max)을 구하여 출력하는 순서도와 슈도코드를 작성하시오.

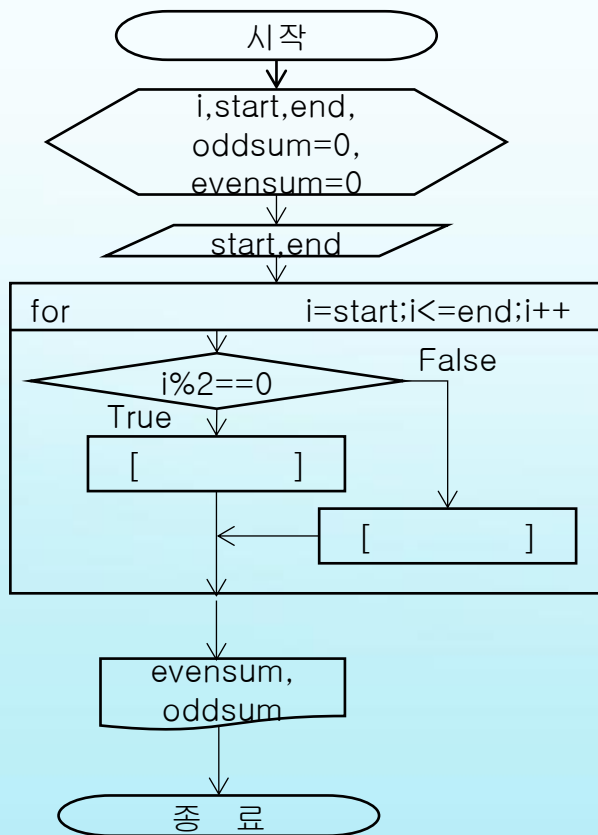
hint

첫 번째 입력 값은 최소값이며 최대값이다.

4.1 일정한 횟수 반복 응용_예제 1

두 수를 입력 받아, 두 수 사이에 홀수의 합과 짝수의 합을 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



pseudocode

```
1  main(){
2      int i, start, end, oddSum=0, evenSum=0;
3      scanf("%d %d", &start, &end);
3      for( i = start; i <= end; i = i + 1 ) {
4          if ( i % 2 == 0 )
5              evenSum = evenSum + i;
6          else
7              oddSum = oddSum + i;
8      }
9      printf("짝수:%d 홀수:%d\n", evenSum, oddSum);
10 }
```

실행 예

```
1 10
짝수:30 홀수:25
```

생각해 보기

1부터 백만까지의 합을 구해야 한다면, if문을 백만 번 수행해야 한다. 보다 효율적인 알고리즘은 있을까?

즉, for문 안에 if문이 있는데, if 문을 for문 밖으로 빼낼 수는 없을까 생각해 보자.

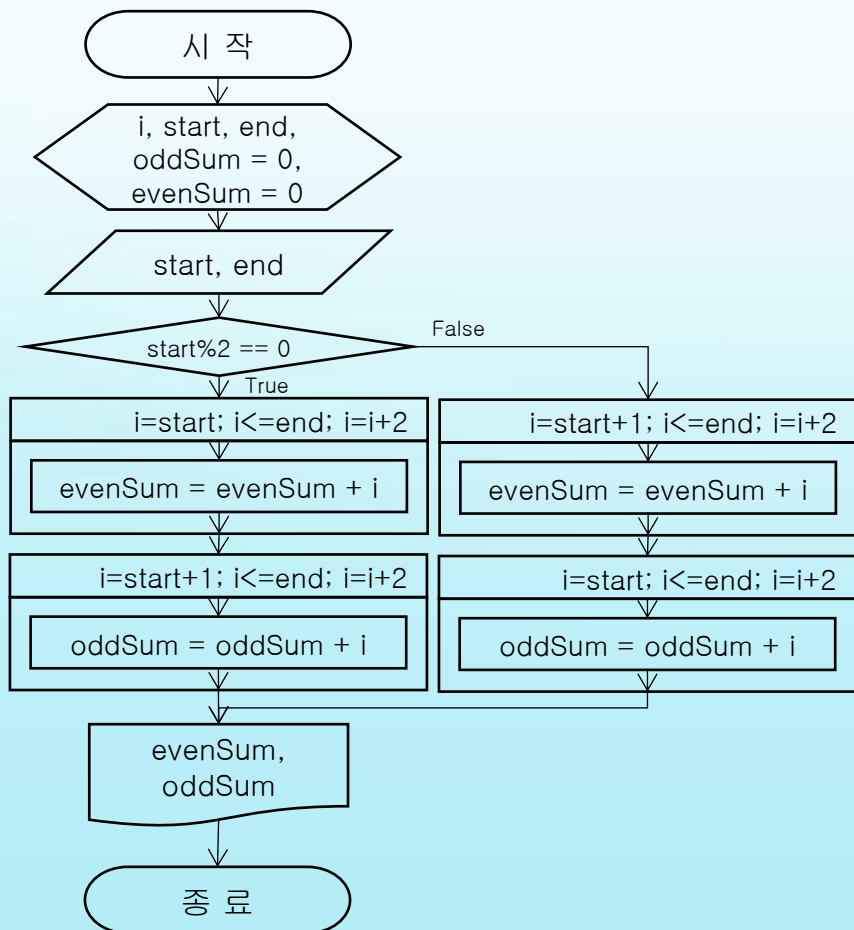
4.1 일정한 횟수 반복 응용 _예제 2

두 수를 입력 받아, 두 수 사이에 홀수의 합과 짝수의 합을 출력하는 순서도와 슈도코드를 작성하시오.

조건

if문을 한번만 수행하도록 하시오.

flowchart



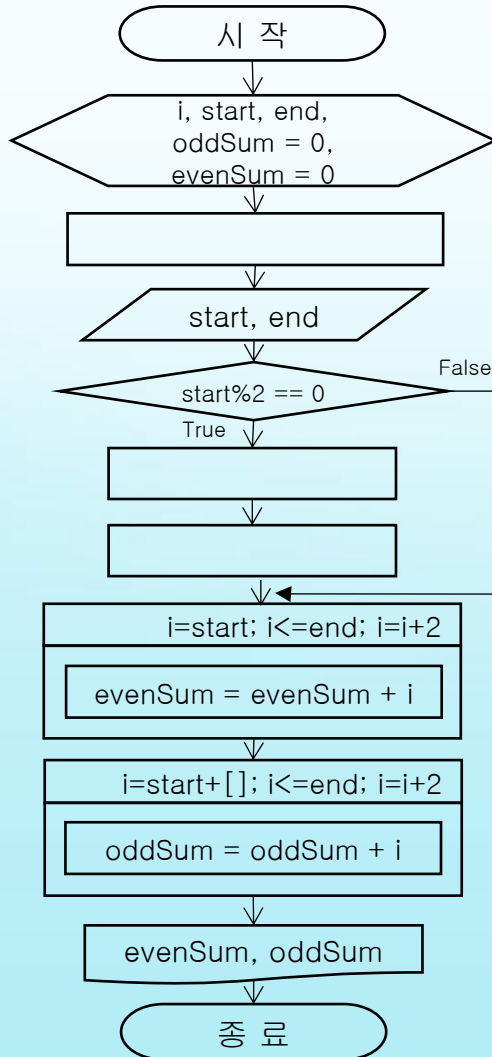
pseudocode

```
1  main(){
2      int i, start, end, oddSum=0, evenSum=0;
3      scanf("%d %d", &start, &end);
4      if (start % 2 == 0) {
5          for( i = start; i <= end; i = i + 2 )
6              evenSum = evenSum + i;
7          for( i = start+1; i <= end; i = i + 2 )
8              oddSum = oddSum + i;
9      }
10     else {
11         for( i = start+1; i <= end; i = i + 2 )
12             evenSum = evenSum + i;
13         for( i = start; i <= end; i = i + 2 )
14             oddSum = oddSum + i;
15     }
16     printf("짝수:%d 홀수:%d\n",
17           evenSum, oddSum);
18 }
```


4.1 일정한 횟수 반복 응용 _문제 1

두 수를 입력받아, 두 수 사이에 홀수의 합과 짝수의 합을 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



조건

for 반복문 코드의 수를 줄이도록 재구성 하시오.

pseudocode

```
1  main(){
2      int i, start, end, oddSum=0, evenSum=0;
3      [
4      scanf("%d %d", &start, &end);
5      if (start % 2 == 0)
6          [
7          [
8      for( i = start+[          ]; i <= end; i = i + 2 )
9          evenSum = evenSum + i;
10     for( i = start+[          ]; i <= end; i = i + 2 )
11         oddSum = oddSum + i;
12     printf("짝수:%d 홀수:%d\n", evenSum, oddSum);
13 }
```

생각해 보기

앞의 수가 큰 경우에는 수행이 안되는데, 이를 해결해보자.

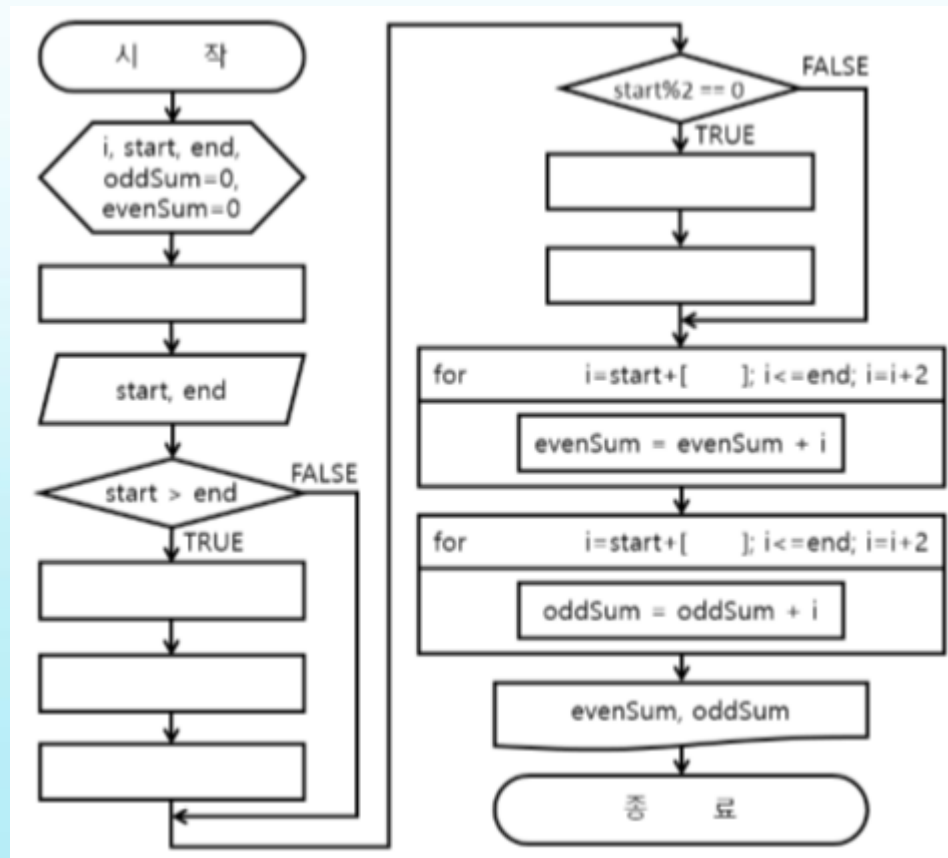
4.1 일정한 횟수 반복 응용 _문제 2

두 수를 입력받아, 두 수 사이에 홀수의 합과 짝수의 합을 출력하는 순서도와 슈도코드를 작성하시오.

조건

수의 크기에 상관없이 수행 가능해야 한다.

flowchart



4.1 일정한 횟수 반복 응용 _문제 2

pseudocode

```
1  main(){
2      int i, start, end, oddSum=0, evenSum=0;
3      [
4      scanf("%d %d", &start, &end);
5      if (start > end) {
6          [
7          [
8          [
9      ]
10     if (start % 2 == 0)
11         [
12         [
13     for( i = start+[
14         evenSum = evenSum + i;
15     for( i = start+[
16         oddSum = oddSum + i;
17     printf("짝수:%d 홀수:%d\n", evenSum, oddSum);
18 }
```

4.1 일정한 횟수 반복_연습 문제 1

다트 게임을 하는 순서도와 슈도코드를 작성하시오.

조건

10X10의 좌표위에 다트의 중심은 난수(random)으로 한다
다트의 크기는 반지름이 2로 한다.

사용자는 X,Y 좌표를 입력하며, 다트 크기 내에 들어가면
맞춘 것으로 하며, 밖에 있을 경우에는 다트 위치에 대한 힌
트를 준다.

몇 번에 맞추었는지 출력하고, 5회까지 못 맞추면 게임을
끝낸다.

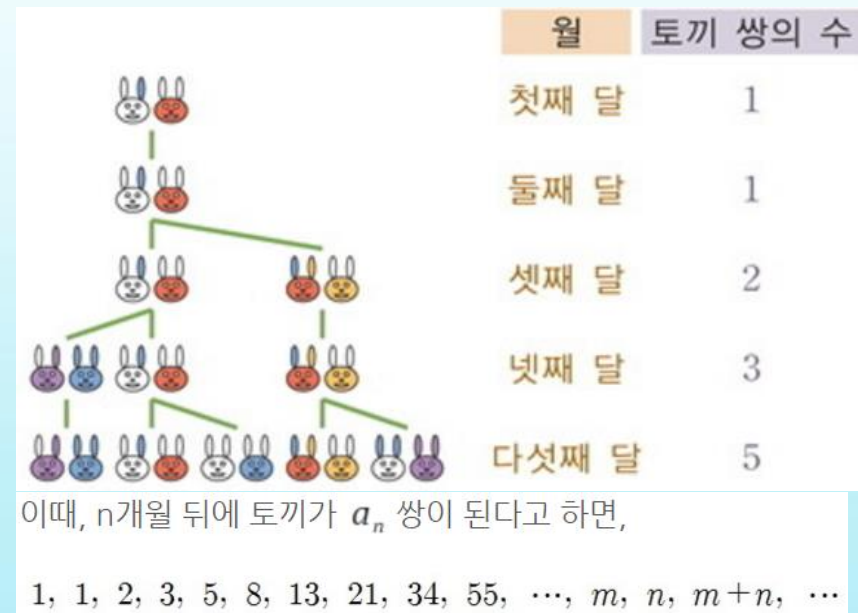


4.1 일정한 횟수 반복_연습 문제 2

한 쌍의 토끼가 있다. 이 토끼는 매달 암수 한 쌍의 새끼를 낳으며, 새로 태어난 토끼도 태어난 지 두 달 후면 어미가 되어 꼭 한 쌍씩의 새끼를 낳는다고 한다. 1년이 지나면 토끼는 모두 몇 쌍이나 태어날까? 1년(12개월) 동안 매달 태어나는 토끼 쌍의 수는 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144이고 합계는 376쌍이다. 연 수를 입력하면 토끼 쌍을 출력하는 순서도와 슈도코드를 작성하시오.

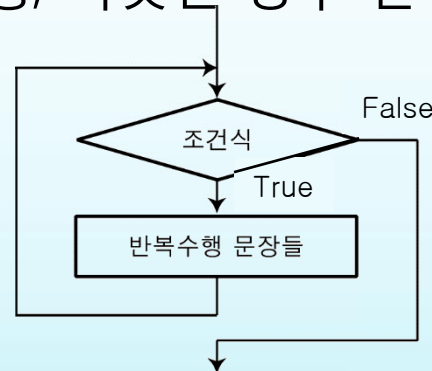
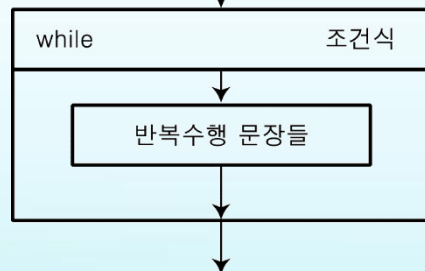
참고

연속된 앞의 두 수의 합이 뒤의 수가 되는 피보나치 수열의 합이다. 자연계에 존재하는 현상이 불가사의한 피보나치 수열과 관계가 있다는 것은 매우 신기한 인연이라고 하겠다.



4.2 선조건 반복

- while 문도 for 문과 똑같이 반복된 수행을 하는 경우 사용됨
- 횟수가 정해져 있는 경우에 사용되는 for 문과는 다르게 while 문은 반복 횟수가 정해져 있는 것이 아니라 어떠한 종료 조건이 주어진 경우 사용
- 조건식이 참인 경우 반복 수행, 거짓인 경우 반복을 벗어남



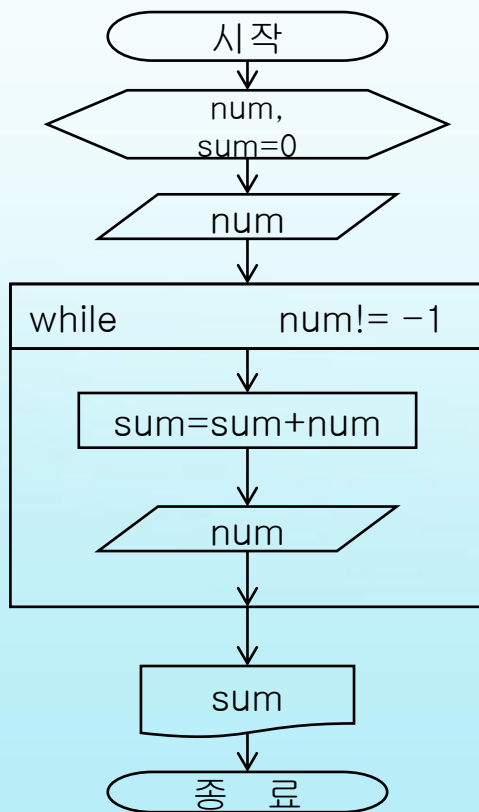
- 특정 조건(어떤 값이 입력)이 될 때까지 반복 수행하는 전형적 구조는 다음과 같다.

```
첫 번째 data 입력;  
while(조건) {  
    반복 수행할 문장들;  
    반복 수행할 문장들;  
    다음 data 입력;  
}
```

4.2 선조건 반복 기초_예제 1

-1 이 입력될 때까지 정수를 입력받아 그 수들의 합을 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



pseudocode

```
1  main(){
2      int num, sum = 0;
3      scanf("%d", &num);
4      while(num != -1)
5      {
6          sum = sum + num;
7          scanf("%d", &num);
8      }
9      printf( " 합은 %d이다.", sum);
10 }
```

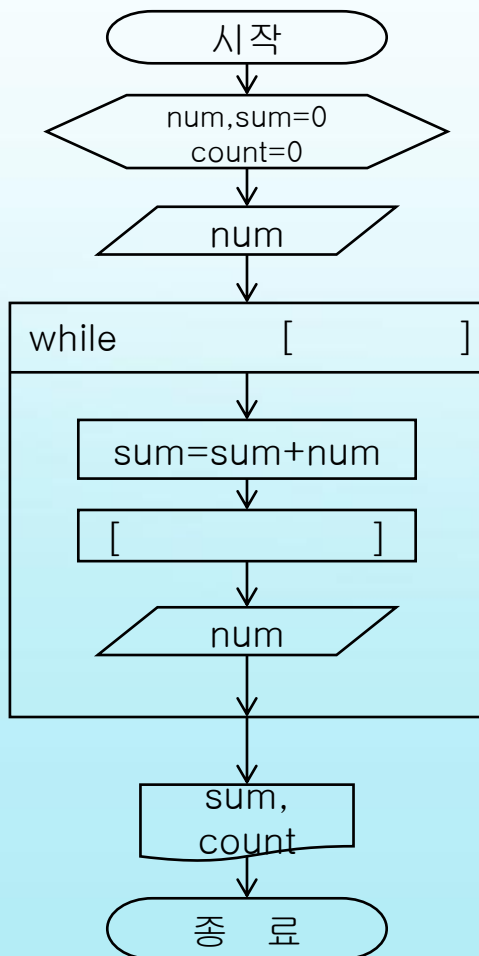
실행 예

```
2
5
4
-1
합은 11이다.
```

4.2 선조건 반복 기초_문제 1

음수가 입력될 때까지, 정수를 입력받아, 그 수들의 합과 입력된 갯수를 출력하는 순서도와 슈도코드를 완성하시오.

flowchart



pseudocode

```
1  main(){
2      int sum = 0, num, count = 0;
3      scanf("%d", &num);
4      while( [ ] )
5      {
6          sum = sum + num;
7          [ ]
8          scanf("%d", &num);
9      }
10     printf("%d %d", sum, count);
11 }
```

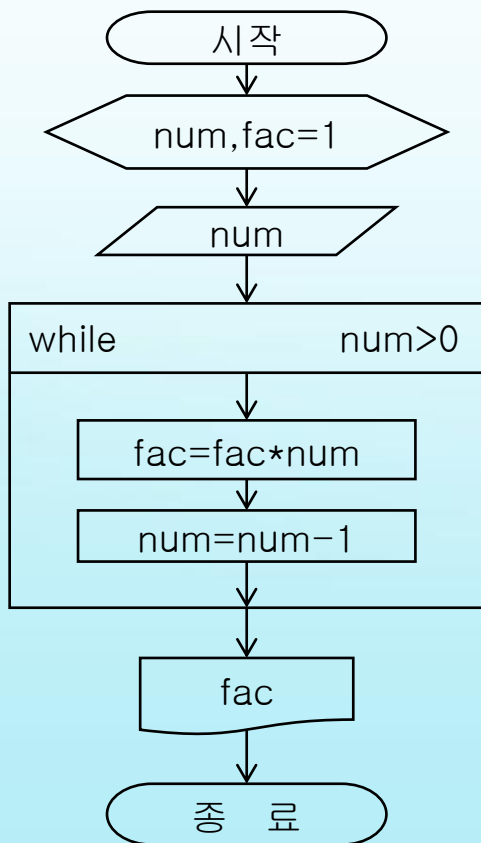
실행 예

```
3
2
5
-1
10 3
```


4.2 선조건 반복 연습_예제 1

어떤 수를 입력받아 그 수의 팩토리얼(Factorial)을 구하는 순서도와 슈도코드를 작성하시오.

flowchart



참고

$n!$ (n 은 양수)은 $1 \sim n$ 까지의 자연수를 모두 곱하는 것
 $n! = 1 * 2 * \dots * (n-1) * n$
 $0!$ (0팩토리얼)은 1이다.

pseudocode

```
1  main(){
2      int num, fac = 1;
3      scanf("%d", &num);
4      while( num > 0) {
5          fac = fac * num;
6          num = num - 1;
7      }
8      printf("%d", fac);
9  }
```

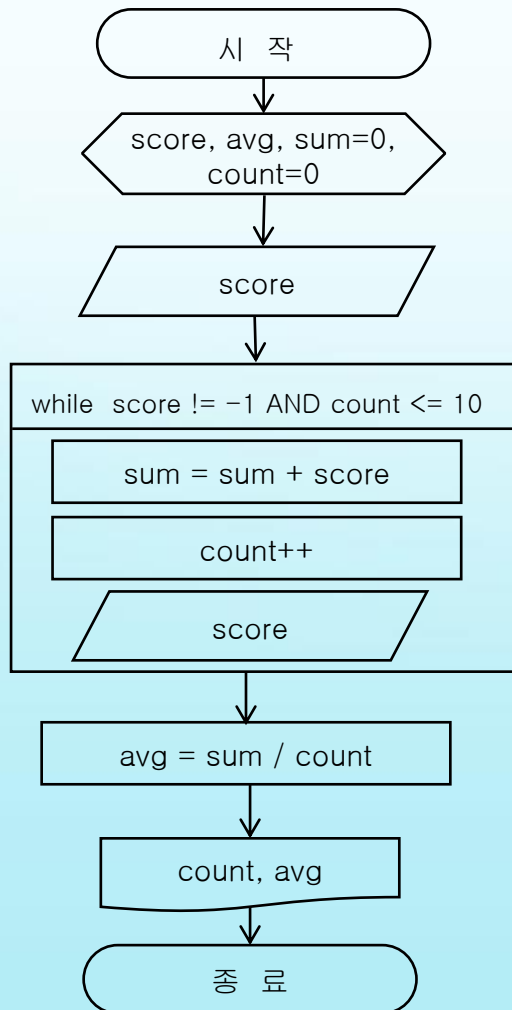
생각해 보기

for문으로 변경해 보자.

4.2 선조건 반복 연습_예제2

성적을 입력받아 학생수와 평균을 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



처리 조건

성적에 -1이 입력되거나 학생 수가 10명이면 입력을 종료

pseudocode

```
1  main(){
2      int score, avg, sum=0, count = 0;
3      scanf("%d", & score);
4      while(score != -1 && count <= 10 ){
5          sum = sum + score;
6          count++;
7          scanf("%d", &score);
8      }
9      avg = sum / count;
9      printf("%d명 학생의 평균은 %d", count, avg);
10 }
```

실행 예

```
80
90
70
-1
3명 학생의 평균은 80
```

4.2 선조건 반복 연습_예제3

나눗셈은 뺄셈으로 부터 시작되었다고 알고 있으나, 나눗셈을 위해 뺄셈이 필요했다. 여러 번의 시행착오를 통해 일정한 수를 연속적으로 빼어서 0을 만들 수 있는 몫을 찾았다. 이런 방법으로 우리 선조들은 농사지어 얻은 수확물과 여러 물건들을 똑같이 나눌 수 있었다. 뺄셈으로 나눗셈을 하는 슈도코드를 작성하시오.

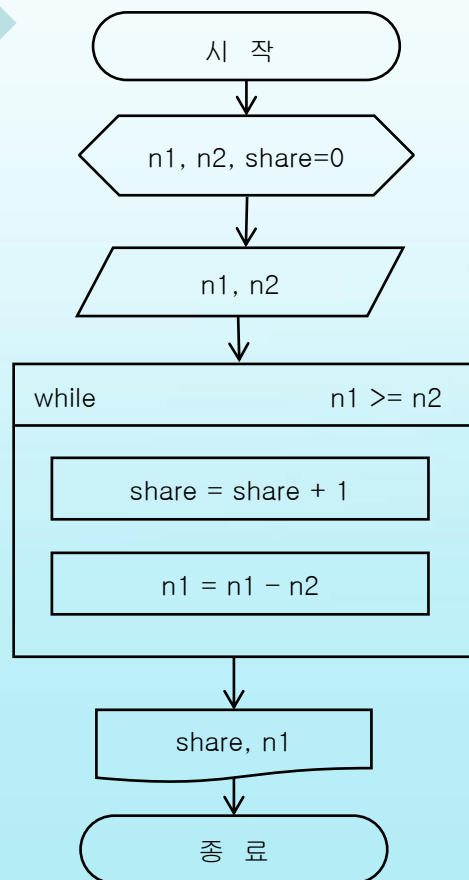
실행 결과

20 5
몫: 4 나머지: 0

생각해보기

나눗셈에서 밑수가 0면 오류가 되는데, 에러 체크를 추가해보자. 또한, 밑수가 0일 경우 위의 슈도코드에서는 어떤 일이 발생할까?

flowchart



4.2 선조건 반복 연습_예제3

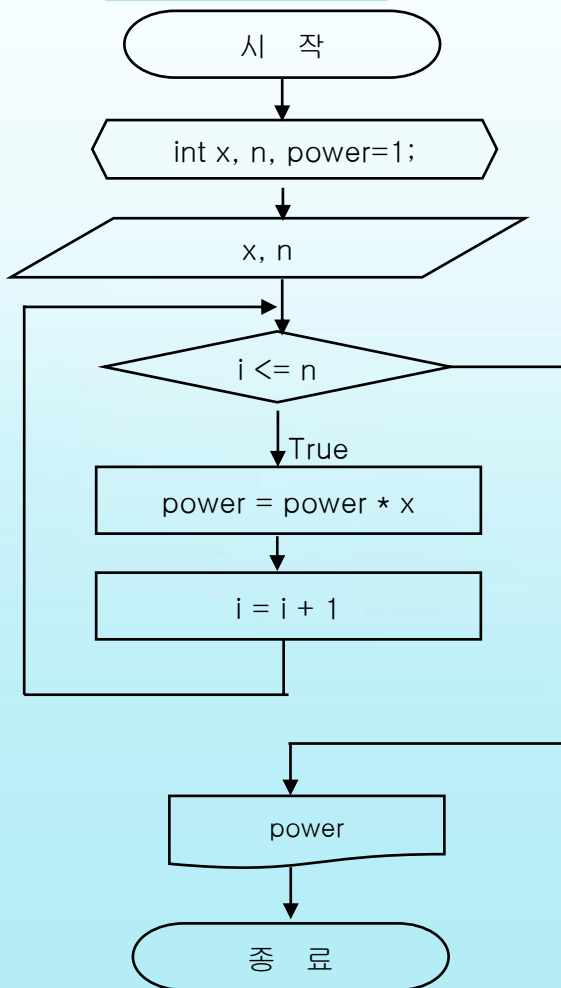
pseudocode

```
1  main() {  
2      int n1, n2, share=0;  
3      scanf("%d %d", &n1, &n2);  
3      while (n1 >= n2 ) {  
4          share = share + 1;  
5          n1 = n1 - n2;  
6      }  
7      printf("몫: %d 나머지:%d\n ", share, n1);  
8  }
```

4.2 선조건 반복 연습_문제 1

양의 정수 n 에 대한 수 x 의 n 제곱을 구하는 알고리즘의 순서도와 슈도코드를 작성하시오.

flowchart



참고

$$X^n = X_1 * X_2 * \dots * X_n$$

$$3^4 = 3 * 3 * 3 * 3$$

C언어에서는 `power()`라는 함수를 이용
`p = power(x, n)`

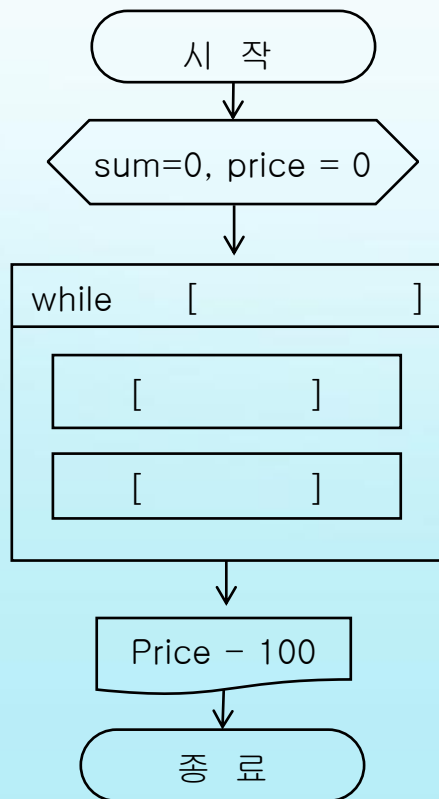
pseudocode

```
1  main(){
2      int i = 1, x, n;
3      int [          ]
4      scanf("%d %d", &x, &n);
5      while( [          ] ) {
6          power = power * x;
7          i = i + 1;
8      }
9      printf("%d", power);
10 }
```

4.2 선조건 반복 연습_문제 2

상품의 가격이 100원 부터 100원 단위로 정해져 있다. 만원을 갖고 있다고 할 때, 살 수 있는 물건의 최대가격을 출력하는 슈도코드를 작성하시오.

flowchart



처리조건

물건은 저렴한 것부터 한 개씩 구입해야 한다

hint

$100+200+300+\dots+N < 10000$

pseudocode

```
1  main(){
2      int sum = 0, price = 0;
4      while( [ ] )
5      {
6          [ ]
7          [ ]
8      }
      printf("%d", price - 100);
11 }
```

4.2 선조건 반복 연습_문제 3

가로가 50cm 세로가 100cm 높이가 50cm인 욕조가 있다. 이 욕조에 1분에 10L씩 물을 채운다고 했을 때 몇 분이 걸리는지 계산하는 알고리즘의 슈도코드를 작성하시오.

참고

1000cc = 1L, 가로60cm 세로 60cm 높이 50cm인 통은 180L이다.

출력 예시

욕조물의 양:10L

욕조물의 양:20L

욕조물의 양:30L

.....

욕조물의 양:250L

욕조에 물이 가득 차는데 25분이 걸립니다.



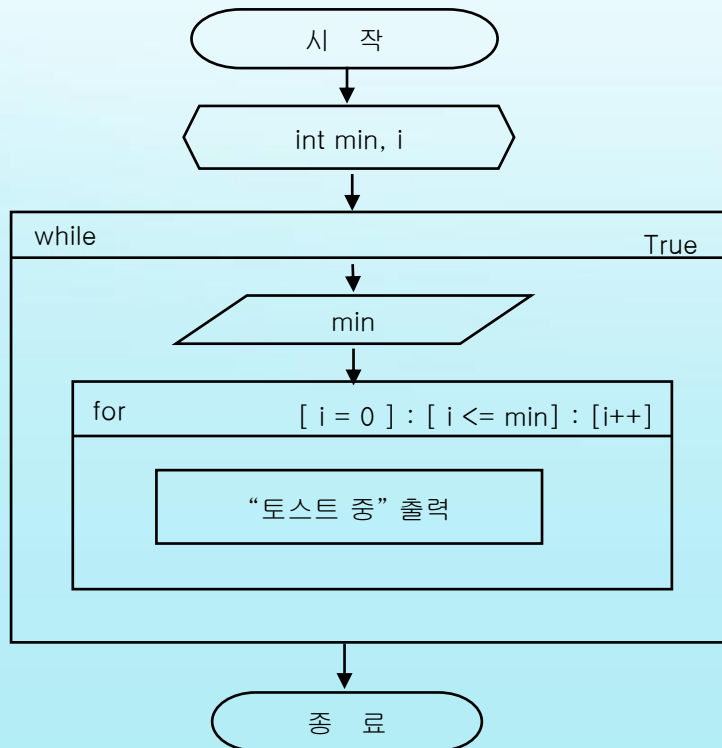
4.2 선조건 반복 응용_예제 1

가전 제품안에 내장된 프로그램의 경우, 전원이 공급되면 프로그램이 무한 반복하면서 버튼등에 입력된 값에 따라 정해진 일을 수행한다. 간단한 빵을 굽는 토스트 안에 프로그램의 순서도와 슈도코드를 작성하시오

조건

1~4분의 선택 버튼이 있다.

flowchart



pseudocode

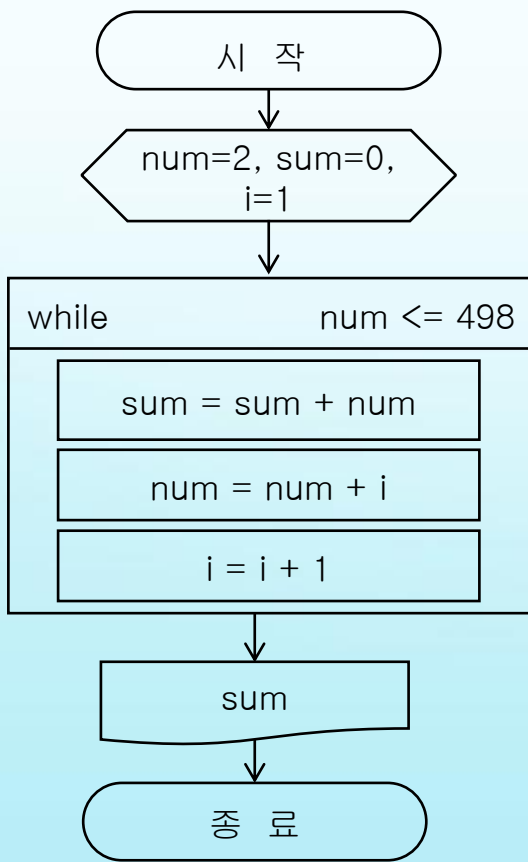
```
1  main(){
2      int min, i;
3      while( 1 )
4      {
5          printf("토스트 시간 입력(1~4)");
6          scanf("%d", min);
7          for(i=0; i<=min; i++)
8              printf("토스팅중\n");
8      }
10 }
```



4.2 선조건 반복 응용_예제 2

2 + 3 + 5 + 8 + 12 + ... + 498의 합을 구하는 순서도와 슈도코드를 작성하시오.

flowchart



참고

어떤 수열이 있을 때 그 수열의 연속적인 두 항의 차를 **계차**라 부르고, 그 계차들을 나열한 것이 계차수열이다.

$$\begin{array}{ccccccccc} 2 & + & 3 & + & 5 & + & 8 & + & 12 & + & \cdots & + & 498 \\ \hline & & 1 & & 2 & & 3 & & 4 & & 5 & & \end{array} \quad \text{계차가 등차수열이다}$$

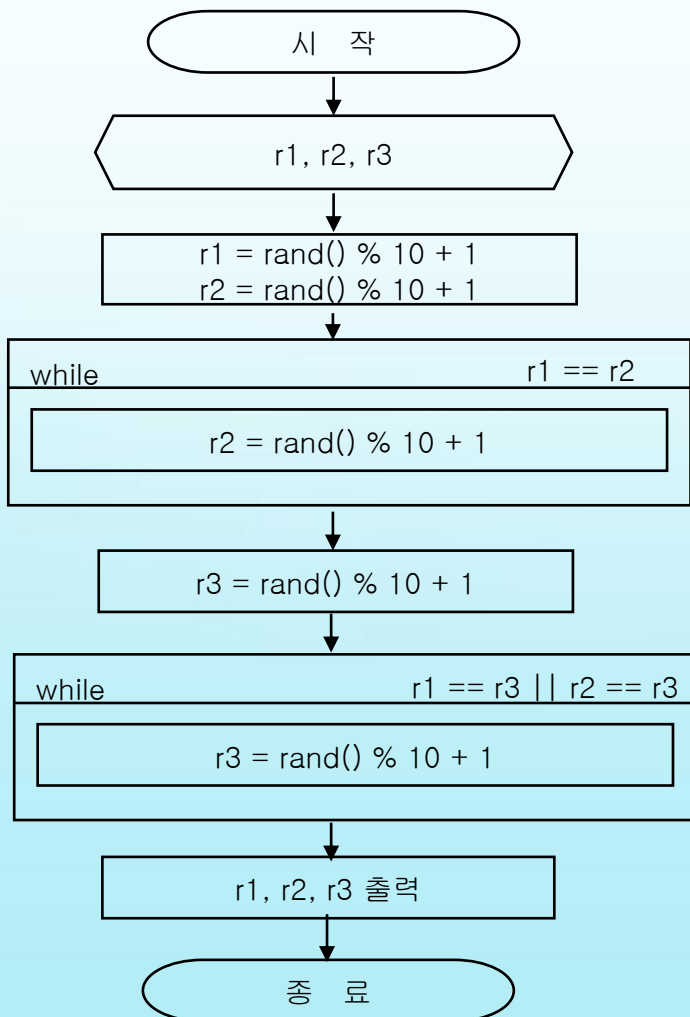
pseudocode

```
1  main(){
2      int num = 2, sum = 0, i = 1;
3      while( num <= 498 )
4      {
5          sum = sum + num;
6          num = num + i;
7          i = i + 1;
8      }
9      printf("%d", sum);
10 }
```

4.2 선조건 반복 응용_예제 3

1-10 사이에 3개의 서로다른 랜덤 값을 출력하는 순서도와 슈도코드를 작성하시오.

flowchart



pseudocode

```
1.  main(){
2.      int r1, r2, r3;
3.      srand((unsigned int)time(NULL));
4.      r1 = rand() % 10 + 1;
5.      r2 = rand() % 10 + 1;
6.      while(r1 == r2){
7.          r2 = rand() % 10 + 1;
8.      }
9.      r3 = rand() % 10 + 1;
10.     while(r1 == r3 || r2 == r3){
11.         r3 = rand() % 10 + 1;
12.     }
13.     printf("%d  %d  %d",r1, r2, r3);
```

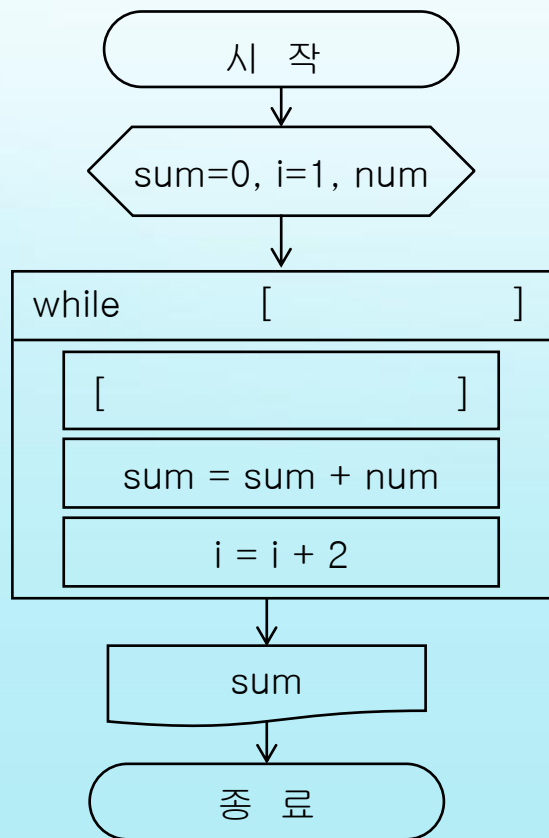
4.2 선조건 반복 응용_문제1

1 + 4 + 9 + 16 + 25 + ... + 2500 의 합을 구하는 알고리즘이다.
빈칸을 채워 완성하시오.

참고

어떤 수열이 있을 때 그 수열의 연속적인 두 항의 차를 **계차**라 부르고, 그 계차들을 나열한 것이 **계차수열**이다.

flowchart



$$\frac{1 + 4 + 9 + 16 + 25 + \dots + 2500}{\quad \quad 3 \quad 5 \quad 7 \quad 9 \quad 11 \quad \quad \quad \text{계차가 등차수열이다}}$$

pseudocode

```
1  main(){
2      int sum = 0, i = 1, num;
3      while( [ ] )
4      {
5          [ ]
6          sum = sum + num;
7          i = i + 2;
8      }
9      printf("%d", sum);
10 }
```

생각해보기

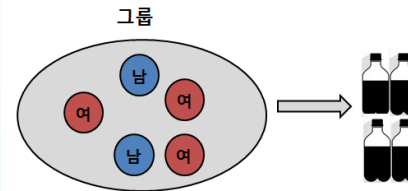
수학적 함수로 만들어 프로그램을 작성해 보시오

4.2 선조건 반복 응용_문제 2

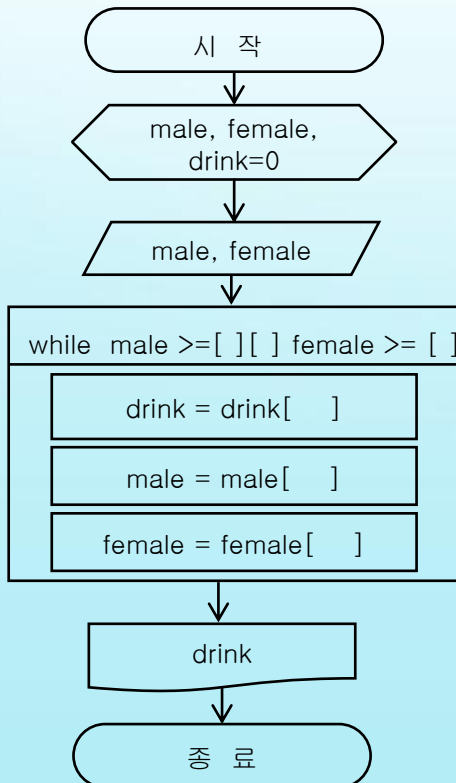
M.T(Membership Training)를 가서 그룹 만들기 게임을 하고있다. 그룹을 만들 때 규칙은 남자 2명과 여자 3명이 한 그룹이 되어야 한다. 그룹 만들기에 성공하면 성공한 그룹마다 음료수를 4개씩 준다. 이 게임에 참가하는 남자와 여자의 인원 수를 입력 시, 필요한 음료수의 개수를 출력하는 코드를 작성하시오.

참고

남 3 여 7 -> drink 4개
남 4 여 7 -> drink 8개
남 4 여 30 -> drink 8개



flowchart



pseudocode

```
1  main() {
2      int male, female, drink=0;
3      scanf("%d %d", &male, &female);
5      while(male>=[ ] [ ] female>=[ ]){
6          drink = drink[ ];
7          male = male[ ];
8          female = female[ ];
9      }
10     printf("음료수는 %d개가 필요하다.", drink);
11 }
```

실행 예

3 7
음료수는 4개가 필요하다.

4.2 선조건 반복 응용_문제 3

입력 받은 십진수를 이진수로 바꿀 때 필요한 이진수의 비트개수를 구하시오.

참고

제곱값을 구하기 위해, pow()를 사용 하시오.
pow(2,3)은 2에 3승으로 $2*2*2=6$ 이 된다



10진수	2진수	비트수	비고
1	1	1	2^0
3	11	2	2^1
7	111	3	2^2
13	1101	4	2^3

pseudocode

```
1  main() {
2      int nbit=0, n;
3      printf("10진수의 값을 입력하시오: ");
4      scanf("%d", &n);
5      while([          ]) {
6          [          ]
7      }
8      printf("\n%d비트가 필요", nbit);
9  }
```

실행 예

10진수의 값을 입력하시오: 3
2비트가 필요

4.2 선조건 반복 응용_문제 4

두 자매가 나이의 비율에 따라 케익을 나누려고 한다. 케익을 가능한 가장 큰 조각으로 나누어(최소 숫자로 나누어) 먹고 싶다. 몇 조각으로 나누어 몇 조각씩 먹으면 되는지를 계산하는 알고리즘의 순서도와 슈도코드를 작성하시오.



힌트

두 나이의 최대공약수(GCD, Greatest Common Divisor)를 찾고, 나이만큼 배분한다.

두 수의 최대공약수는 두 수가 같을 때까지 큰 수에서 작은 수를 빼나가면 된다.

10살과 15살 이라면, 최대공약수는 5이다.

케익을 5조각으로 나누고, 10살은 2조각, 15살은 3조각을 먹으면 된다.

최대공약수가 없으면(즉 1이면), 두 나이의 합만큼 나누어 나이만큼 조각을 갖는다.

pseudocode

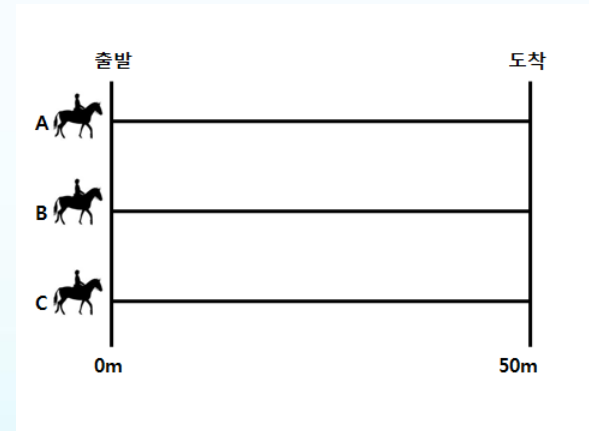
```
1  main() {
2      int age1, age2;
3      int gcd1, gcd2;
4      int portion1, portion2;
5      scanf("%d %d", &age1, &age2);
6      gcd1 = age1;
7      gcd2 = age2;
8      while (gcd1 != gcd2) {
9          if (gcd1 > gcd2)
10             gcd1 = gcd1 - gcd2;
11         else
12             gcd2 = gcd2 - gcd1;
13     }
14     printf("GCD of %d and %d is %d\n", age1, age2, gcd1);
15     portion1 = age1 / gcd1;
16     portion2 = age2 / gcd1;
17     printf("Cake should be divided into %d\n", portion1+portion2);
18     printf("The %d year old gets %d and the %d year old gets %d\n",
19         age1, portion1, age2, portion2);
20 }
```

4.2 선조건 반복_연습 문제 1

경마게임으로, 경주마를 3필로 하고, 50m를 경주하는 프로그램을 작성하시오.

처리 조건

말은 `rand()%10`으로 한번에 0~9까지 가게 하시오.
 말들이 달렸을 때마다 말의 위치를 m로 출력하시오.
 말이 결승점에 도착하면 말의 위치를 50m로 표시하시오.
 말이 동시에 2필 이상 도착 시 알려주고 종료 시키시오.



출력 예시

```
A말 = 1m B = 3m C말 = 1m
A말 = 1m B = 6m C말 = 4m
A말 = 1m B = 8m C말 = 8m
A말 = 3m B = 16m C말 = 10m
A말 = 5m B = 20m C말 = 14m
A말 = 11m B = 26m C말 = 17m
A말 = 11m B = 29m C말 = 21m
A말 = 16m B = 29m C말 = 23m
A말 = 24m B = 36m C말 = 25m
A말 = 24m B = 41m C말 = 25m
A말 = 32m B = 43m C말 = 26m
A말 = 39m B = 45m C말 = 32m
A말 = 42m B = 47m C말 = 37m
A말 도착!!
B말 도착!!
A말 = 50m B = 50m C말 = 43m
말 동시 도착
```

```
A말 = 0m B = 4m C말 = 2m
A말 = 3m B = 9m C말 = 4m
A말 = 10m B = 15m C말 = 5m
A말 = 13m B = 19m C말 = 11m
A말 = 13m B = 22m C말 = 14m
A말 = 13m B = 26m C말 = 22m
A말 = 17m B = 27m C말 = 25m
A말 = 20m B = 27m C말 = 25m
A말 = 27m B = 32m C말 = 30m
A말 = 30m B = 36m C말 = 37m
A말 = 31m B = 44m C말 = 40m
A말 = 31m B = 49m C말 = 44m
B말 도착!!
A말 = 31m B = 50m C말 = 49m
경기 종료
```


4.2 선조건 반복_연습 문제 2

첫 번째 날(D1)에 10원을 예금하고, 그 다음 날에는 전날의 2배를 예금하기를 계속할 경우, 예금 총액(S1)이 10000원이 넘는 날과 총 예금액 및 하루 평균 예금액(M1)을 구하는 프로그램의 순서도와 슈도코드를 작성하시오.

참고

일자(D1)	예금액(B1)	총예금액(S1)
1	10	10
2	20	30
3	40	70
4	80	150
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	10000 이상

4.2 선조건 반복_연습 문제 3

두 수(num1, num2)의 최대공약수와 최소공배수를 구하여라.

hint

- ① 최대공약수 : 공약수(두 개 이상의 자연수의 공통인 약수) 중에서 가장 큰 수예를 들어, 8의 약수는 1, 2, 4, 8이며, 12의 약수는 1, 2, 3, 4, 6, 12이다. 따라서 8과 12의 공약수는 1, 2, 4이며, 최대공약수는 4이다.
- ② 최소공배수 : 공배수(두 개 이상의 자연수의 공통인 배수) 중 가장 작은 수예를 들어, 4의 배수는 4, 8, 12, 16, 20, 24, ... 이며, 6의 배수는 6, 12, 18, 24, ... 이다. 따라서 4와 6의 공배수는 12, 24, ... 이며, 최소공배수는 12이다.

4.2 선조건 반복_연습 문제 4

컴퓨터가 난수(random number)를 한 개 정해주면 그 수(0~9)를 알아 맞추는 게임의 순서도와 슈도코드를 작성하시오. 사용자가 한 수를 입력하면 그 정해진 수보다 큰지 작은지를 알려주어 몇 번 만에 맞추는지 출력한다

참고

컴퓨터 난수 생성 함수는 `rand()`이며, 1~32767까지의 수를 발생
0~9까지의 난수 발생은 `num = rand()%10` 으로 한다.

실행 예

컴퓨터의 수는? 2
2보다 큰 수입니다.
컴퓨터의 수는? 4
4보다 큰 수입니다.
컴퓨터의 수는? 6
6보다 큰 수입니다.
컴퓨터의 수는? 9
9보다 작은 수입니다.
컴퓨터의 수는? 8
8보다 작은 수입니다.
컴퓨터의 수는? 7
정답입니다. 6번째 맞추었습니다.

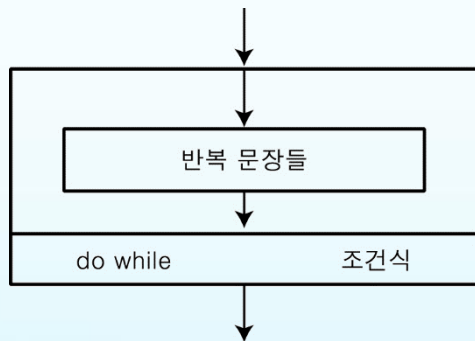


4.2 선조건 반복_연습 문제 5

냉장고는 전원이 공급되면, 내장되어 있는 프로그램이 동작한다. 실제 냉장고의 사용자 인터페이스를 참고해서 동작하는 프로그램의 순서도와 슈도코드를 작성하시오.

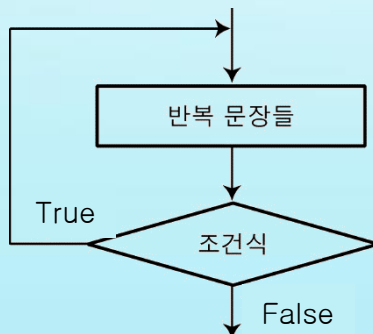
4.3 후조건 반복_1

- do - while 문은 조건 보다 먼저 수행
- 수행 후에 조건을 검사하여 참인 동안은 반복 문장을 계속 수행하고 거짓이면 do - while 문을 벗어남



- ① 반복 문장들을 수행
- ② 조건식을 검사
- ③ ②에서 조건식이 참이면 1부터 다시 실행
- ④ ②에서 조건식이 거짓이면 do - while 문을 벗어남

- if 문 표현



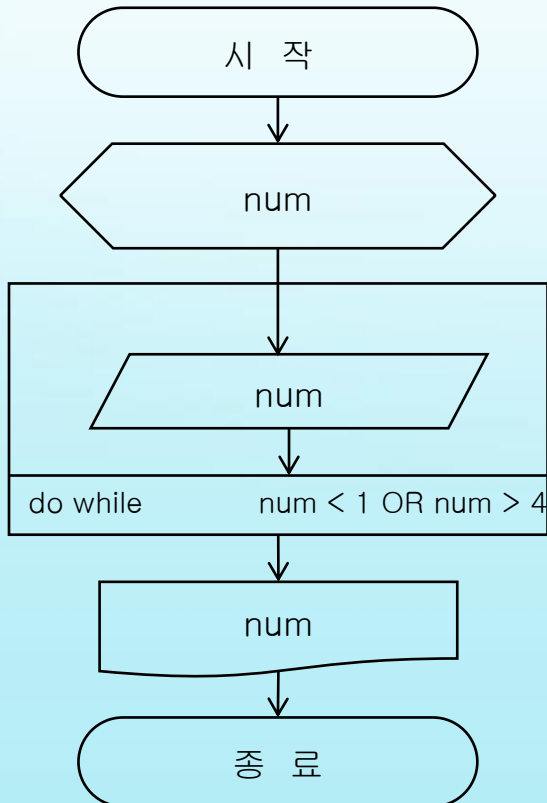
4.3 후조건 반복 기초_예제 1

1부터 4까지의 수만을 입력받아 해당 수를 출력하는 프로그램이다. 순서도와 슈도코드를 작성하시오.

처리조건

1부터 4 이외의 수를 입력받으면 입력을 다시 받음

flowchart



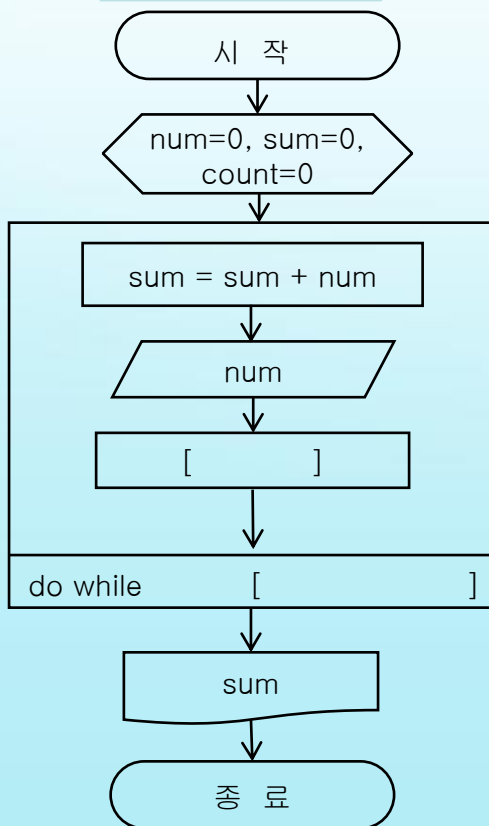
pseudocode

```
1  main(){
2      int num = 0;
3      do{
4          scanf("%d", &num);
5      }while( num < 1 || num > 4 );
6      printf("%d", num);
7  }
```

4.3 후조건 반복 기초_문제 1

어떤 수들을 입력받아 그 수들의 합을 출력하는 순서도와 슈도코드를 작성하시오. 이문제를 do - while 문을 사용하여 작성하시오.(단, -1이 입력되거나 입력된 수가 10개이면 입력을 종료한다.)

flowchart



pseudocode

```
1  main(){
2      int num=0, sum=0, count=0
3      do{
4          sum = sum + num
5          scanf("%d", &num);
6          [ ]
7      }while([ ]);
8      printf("%d", sum);
9 }
```

4.3 후조건 반복 기초_문제 2

피연산자 두 개를 먼저 입력받고 메뉴에 있는 연산자를 선택하여 사칙연산을 하는 프로그램의 순서도와 슈도코드를 작성하여라.

- ① 메뉴에서 1~5사이가 아니면 다시 입력
- ② END가 선택될 때까지 반복

메뉴항목

1. +
2. -
3. *
4. /

실행예시

피연산자 1 : 34
피연산자 2 : 12
1. +
2. -
3. *
4. /
연산자를 선택하시오 : 2
34 - 12 = 22

4.3 후조건 반복 기초_문제 3

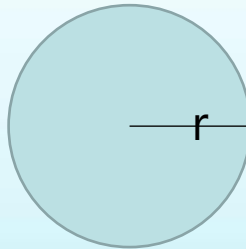
한 번의 길이를 입력받고 원의 넓이, 정삼각형, 정사각형, 정육각형의 넓이 중 하나를 선택하여 계산하는 순서도와 슈도코드를 작성하시오.

- ① 5가 선택될 때까지 반복
- ② 1~5가 아니면 "error" 출력 후 다시 입력

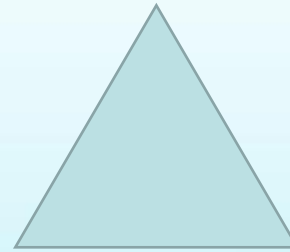
메뉴항목

- 1 : 원의 넓이
- 2 : 정삼각형의 넓이
- 3 : 정사각형의 넓이
- 4 : 정육각형의 넓이
- 5 : 종 료

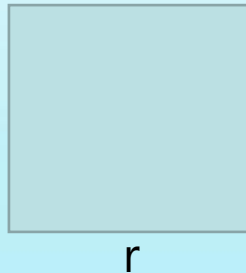
1.



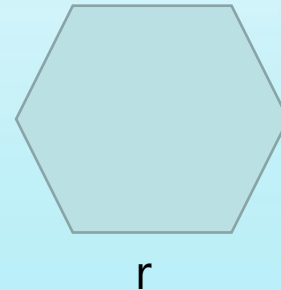
2.



3.



4.

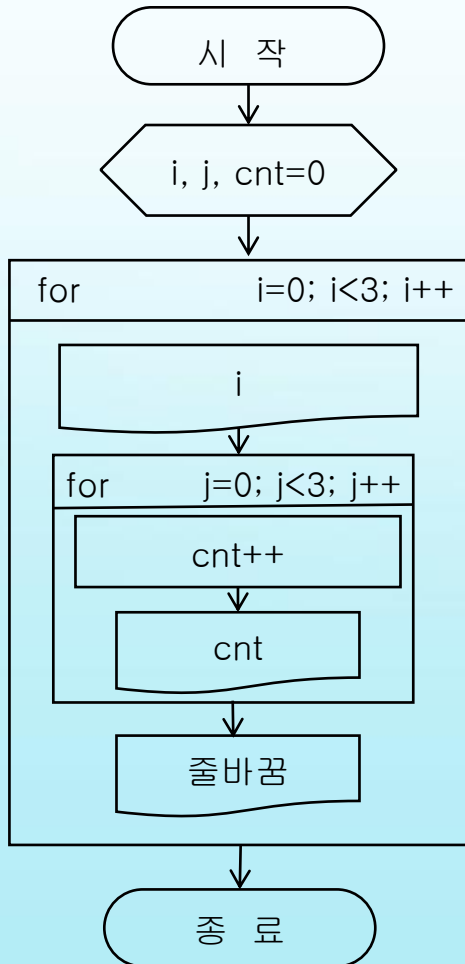


4.4 중첩 반복_예제1

for 문을 중첩해서 다음의 결과처럼 출력하는 순서도와 슈도코드를 완성하시오.

```
i = 0 1 2 3  
i = 1 4 5 6  
i = 2 7 8 9
```

flowchart



pseudocode

```
1  main(){  
3      int i, j, cnt = 0;  
4      for(i = 0; i < 3; i++){  
5          printf("i = %d ", i);  
6          for(j = 0; j < 3; j++){  
7              cnt ++;  
8              printf("%d ", cnt);  
9          }  
10         printf("\n");  
11     }  
12 }
```



4.4 중첩 반복_문제 1

입력 숫자만큼 라인 수를 입력 받아 *로 다음 조건 모양을 출력하는
순서도와 슈도코드를 작성하시오.

① 직각삼각형 모양

1 *
2 **
3 ***
4 ****
5 *****

② 정삼각형 모양

1 * 1
2 *** 3
3 ***** 5
4 ******** 7
5 ********** 9

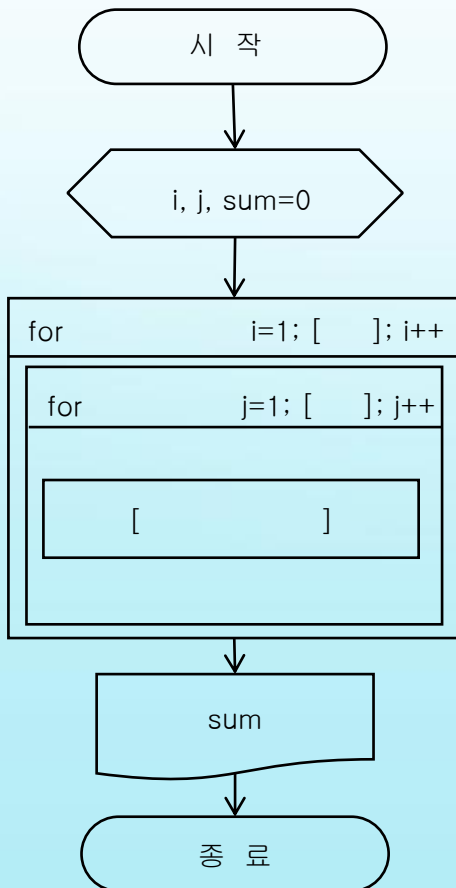
3 마름모 모양

1 * 1
2 *** 3
3 ***** 5
4 *** 3
5 * 1

4.4 중첩 반복_문제 2

sum = 1 + (1+2) + (1+2+3) + ... + (1+2+3+...+100)의 합을 출력하기 위한 순서도와 슈도코드를 작성하시오.

flowchart



pseudocode

```
1  main(){
3      int i, j, sum = 0;
4      for(i = 1; [ ]; i++){
5          for(j = 1; [ ]; j++){
6              [ ]
7          }
8      }
9      printf("%d", sum);
10 }
```

생각해 보기

중첩 for문을 사용하지 않고 보다 효율적인 알고리즘을 생각해 보자



4.4 중첩 반복_문제 3

다음 형식의 타임테이블을 출력하는 알고리즘의 슈도코드를 작성하시오.

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	19	20
3	3	6	9	12	15	18	21	24	27	30
:										
10	10	20	30	40	50	60	70	80	90	100

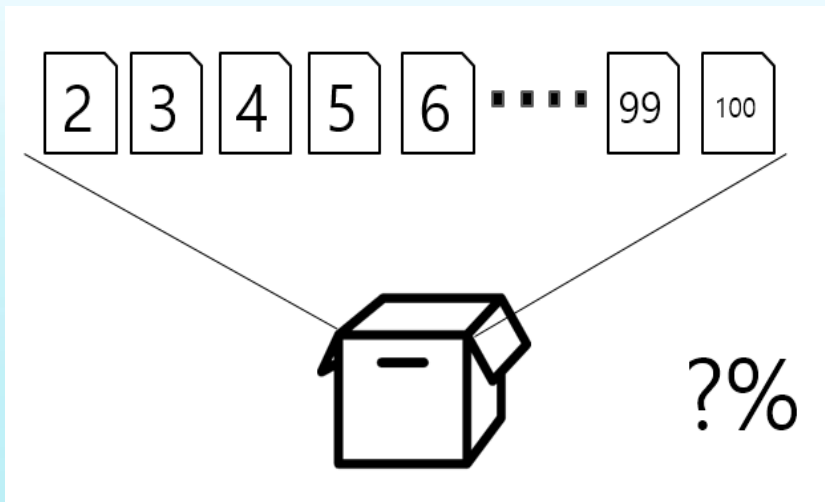


4.4 중첩 반복_문제 4

5에서부터 소수에 해당되는 수 10개를 찾아서 그 합을 구하는 순서도와 슈도코드를 작성하시오(소수란 1과 자기 자신 외에는 더 이상 나누어 떨어지지 않는 수로 1은 제외한다).

4.4 중첩 반복_연습 문제 1

2부터 100까지의 숫자가 각각 적혀있는 공이든 상자가 있다. 이 공 중에서 소수의 숫자가 적힌 종이가 '꽝'일 때 상자에서 공을 한 개 꺼낼 시 꽝일 확률을 구하시오.



실행 예

2부터 100까지의 소수는 25개이다.
꽝을 뽑을 확률은 25%이다.

4.4 중첩 반복_연습 문제 2

추의 종류가 2g, 3g, 5g이 각각 10개씩 있다. 이 추들을 이용하여 전체 무게가 81g인 추의 조합을 찾아, 추의 개수를 출력하는 프로그램의 순서도와 슈도코드를 작성하시오.

hint

문제를 수식으로 표현하면,

$$2x + 3y + 5z = 81$$

$$1 \leq x \leq 10$$

$$1 \leq y \leq 10$$

$$1 \leq z \leq 10$$

출력예시

2 gram	3 gram	5 gram
8	10	7
10	7	8
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.

4.4 중첩 반복_연습 문제 3

다음의 계산식에 맞는 모든 경우의 A, B, C, D, E를 구하는 프로그램의 순서도와 슈도코드를 작성하시오.

출력 예시

- ① A, B, C, D, E는 서로 다른 정수이고, 한 자리 정수(0에서 6까지)이다.
- ② 같은 경우의 수가 2번 이상 출력되지 않는다.
- ③ 전체 출력되는 경우는 24가지이다.
- ④ 다음의 예와 같이 출력한다.

A = 0, B = 3, C = 6, D = 1, E = 5일 경우
0 3 6 1 + 6 3 0 5 = 6666이 된다.