


# Loops

# While Loops

```
int i = 0;
while (i < 10)
{
    printf("%d\n", i);
    i = i + 1;
}
```

loop condition



Loops (repeats body) as long as condition is true.

# For Loops

```
//test
int i;
for (i = 0; i < 10; ++i)
{
    printf("%d\n", i);
}
```

# Looping Through Array

```
//test
int i;
int items[] = { 0,1,1,2,3,5,8,13 };
int itemCount = sizeof(items)/sizeof(int);

for (i = 0; i < itemCount; ++i)
{
    printf("%d\n",items[i]);
}
```

# Accumulate Pattern

```
int i,total = 0;
for (i = 0; i < itemCount; ++i)
{
    total += items[i];    //equivalent to total = total + items[i];
}
//total now holds the sum of all items
```

# Search Pattern

```
int
find(int target,int *items,int itemCount)
{
    int i;
    for (i = 0; i < itemCount; ++i)
    {
        if (items[i] == target)
        {
            return 1;           // short-circuit! return true
        }
    }
    return 0;    //must not be in the array, return false
}
```

# Filter Pattern

```
void findMonthTweets(Tweet **tweets, int count, int month)
{
    int index = 0;
    while (index < count)
    {
        Tweet *tweet = tweets[index];
        if (tweet->month == month)
            printf("%s %s %d %d\n", tweet->user, tweet->tweetBody, tweet->month, tweet->day);
        index++;
    }
}
```

# Extreme Pattern

```
void findLatestTweet(Tweet **tweets, int count)
{
    int index = 1;
    Tweet *latestTweet = tweets[0];
    while (index < count)
    {
        Tweet *tweet = tweets[index];
        if (tweet->month > latestTweet->month)
            latestTweet = tweet;
        else if (tweet->month == latestTweet->month && tweet->day > latestTweet->day)
            latestTweet = tweet;
        index++;
    }
    printf("%s %s %d %d\n", latestTweet->user, latestTweet->tweetBody, latestTweet->month, latestTweet->day);
}
```



# Infinite Loops

In fact, a while loop of the general form:

```
i = INIT;  
while (i < LIMIT)  
{  
    // body  
    ...  
    i += STEP;  
}
```

can be written as for loop:

```
for (i = INIT; i < LIMIT; i += STEP)  
{  
    // body  
    ...  
}
```

# Convert

```
int total = 0;
int j = 1;
while (j < 20)
{
    if (j % 2 == 0) total += 1;
    j++;
}
```

```
int product = 1;
int k;
for (k = 0; k < 30; k++)
{
    if (k % 2 != 0) product *= k;
}

return 0;
}
```