



# Example

- Calculate the sum of the following series ( $x$  and  $m$  are user inputs).

$$x^0 + x^1 + x^2 + x^3 + \dots + x^m$$

# Example

- Calculate the sum of the following series ( $x$  and  $m$  are user inputs).

$$x^0 + x^1 + x^2 + x^3 + \dots + x^m$$

```
#include <stdio.h>
int main()
{
    int x, m, term = 1, sum = 1;
    scanf("%d%d", &x, &m);
    for(i=1; i<=m; i++)
    {
        term = term * x;
        sum = sum + term;
    }
    printf("Sum = %lf", sum);
    return 0;
}
```

# Selection Inside Loop

- Read a positive integer and determine if it is a prime number.
- Pseudo-code:
  - Read integer and store it in ***number***
  - Set ***flag*** to 1
  - For ***i*** = 2 to (***number***-1)
  - If ***number*** is divisible by ***i*** then
    - Set ***flag*** to 0
  - If ***flag*** equals 1, then the number is a prime number, otherwise not

# Selection Inside Loop

- Read a positive integer and determine if it is a prime number.

```
#include <stdio.h>
void main()
{
    int number, i, flag = 1;
    scanf("%d", &number);
    for(i=2; i<number; i++)
    {
        if(number % i == 0)
            flag = 0;
    }
    if(flag == 1)
        printf("%d is a prime number", number);
    else
        printf("%d is not a prime number", number);
}
```

# Example

- Calculate the sum of the following series ( $n$  is user input).

$$\frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} - \dots \pm \frac{1}{n}$$

# Example

- Calculate the sum of the following series ( $n$  is user input).

$$\frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} - \dots \pm \frac{1}{n}$$

```
#include <stdio.h>
int main()
{
    int n, i;
    double term, sum = 0;
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        term = 1.0 / i;
        if(i%2 == 0)
            sum = sum - term;
        else
            sum = sum + term;
    }
    printf("Sum = %lf", sum);
    return 0;
}
```

# Using **break** Inside Loop

- In the prime number example, we do not need to continue the loop till the end once the value of flag is set to zero.

```
#include <stdio.h>
int main()
{
    int number, i, flag = 1;
    scanf("%d", &number);
    for(i=2; i<number; i++)
    {
        if(number % i == 0)
        {
            flag = 0;
            break;
        }
    }
    if(flag == 1)
        printf("%d is a prime number", number);
    else printf("%d is not a prime number", number);
    return 0;
}
```

The **break** statement makes the loop terminate prematurely.



# Using **continue** Inside Loop

- Read 10 integers from the user and calculate the sum of the positive numbers.

```
#include <stdio.h>
int main()
{
    int number, i, sum = 0;

    for(i=0; i<10; i++)
    {
        printf("Enter a number: ");
        scanf("%d", &number);
        if(number < 0)
            continue;
        sum = sum + number;
        printf("%d is added\n", number);
    }
    printf("Total = %d", sum);
    return 0;
}
```

The **continue** statement forces next iteration of the loop, skipping any remaining statements in the loop

# Using `continue` Inside Loop

- Read 10 integers from the user and calculate the sum of the positive numbers.

```
#include <stdio.h>
int main()
{
    int number, i, sum = 0;

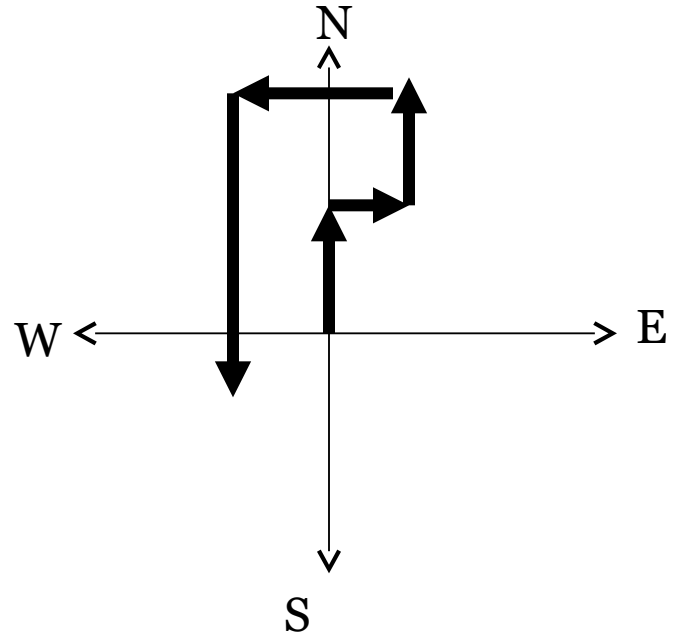
    for(i=0; i<6; i++)
    {
        printf("Enter a number: ");
        scanf("%d", &number);
        if(number < 0)
            continue;
        sum = sum + number;
        printf("%d is added\n", number);
    }
    printf("Total = %d", sum);
    return 0;
}
```

## **Output:**

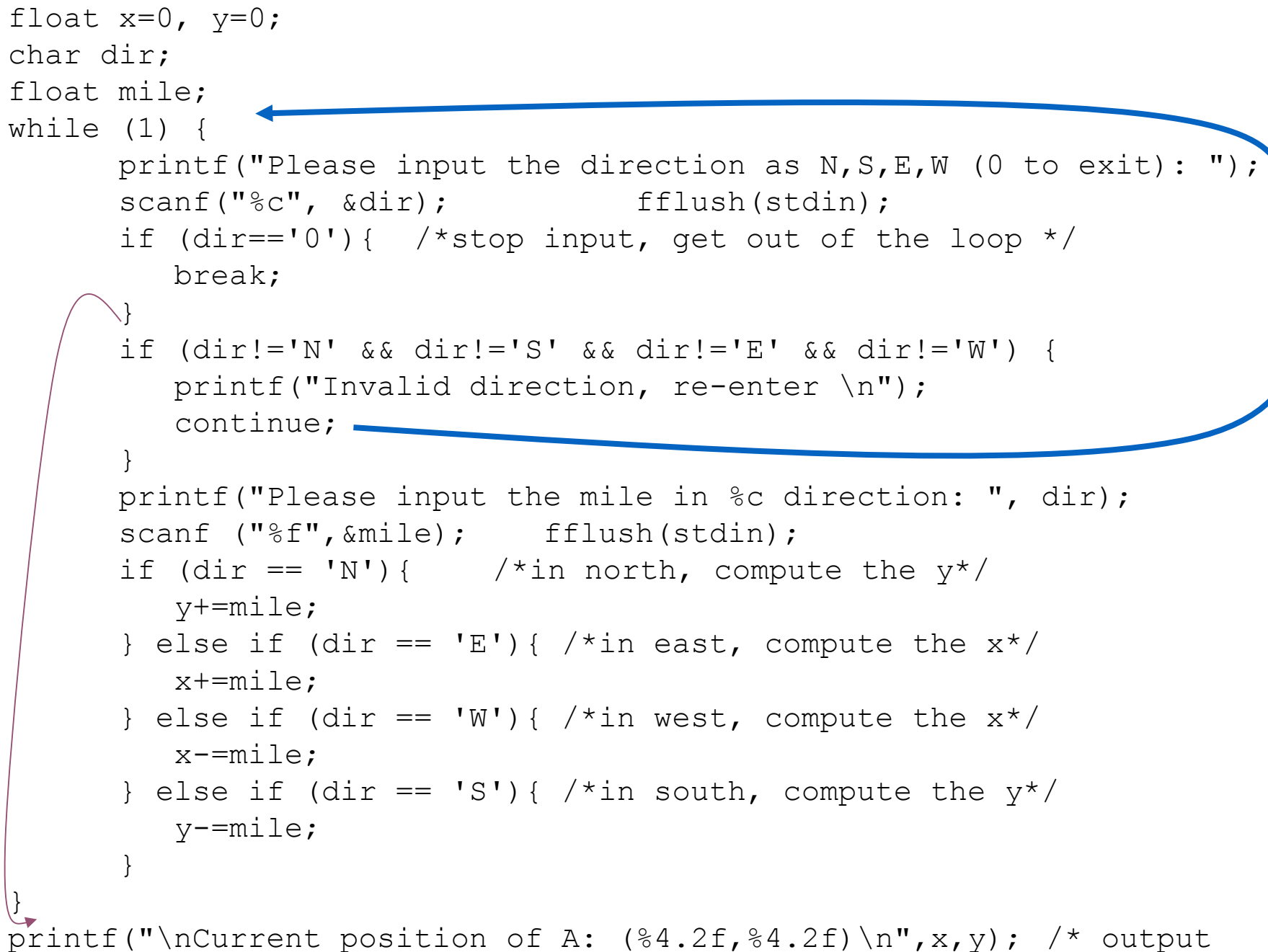
```
Enter a number:
1
1 is added
Enter a number:
2
2 is added
Enter a number:
3
3 is added
Enter a number:
-4
Enter a number:
-5
Enter a number:
6
6 is added
Total = 12
```

# Example: A Travelling Man

- Suppose a man (say, A) stands at (0, 0) and waits for user to give him the direction and distance to go.
- User may enter N E W S for north, east, west, south, and any value for distance.
- When user enters 0 as direction, stop and print out the location where the man stopped



```
float x=0, y=0;
char dir;
float mile;
while (1) {
    printf("Please input the direction as N,S,E,W (0 to exit): ");
    scanf("%c", &dir);          fflush(stdin);
    if (dir=='0'){ /*stop input, get out of the loop */
        break;
    }
    if (dir!='N' && dir!='S' && dir!='E' && dir!='W') {
        printf("Invalid direction, re-enter \n");
        continue;
    }
    printf("Please input the mile in %c direction: ", dir);
    scanf ("%f",&mile);      fflush(stdin);
    if (dir == 'N'){ /*in north, compute the y*/
        y+=mile;
    } else if (dir == 'E'){ /*in east, compute the x*/
        x+=mile;
    } else if (dir == 'W'){ /*in west, compute the x*/
        x-=mile;
    } else if (dir == 'S'){ /*in south, compute the y*/
        y-=mile;
    }
}
printf("\nCurrent position of A: (%4.2f,%4.2f)\n",x,y); /* output
```



# Nested Loop

- What is the output of the following program?

```
for (i=1; i<=5; i++)
{
    for (j=1; j<=4; j++)
    {
        printf("*");
    }
    printf("\n");
}
```

# Nested Loop

- What is the output of the following program?

```
for (i=1; i<=5; i++)  
{  
    for (j=1; j<=4; j++)  
    {  
        printf("*");  
    }  
    printf("\n");  
}
```

Output

```
****  
****  
****  
****  
****
```

# Nested Loop

- What is the output of the following program?

```
for (i=1; i<=5; i++)
{
    for (j=1; j<=i; j++)
    {
        printf("*");
    }
    printf("\n");
}
```

# Nested Loop

- What is the output of the following program?

```
for (i=1; i<=5; i++)  
{  
    for (j=1; j<=i; j++)  
    {  
        printf("*");  
    }  
    printf("\n");  
}
```

Output

```
*  
**  
***  
****  
*****
```



# Nested Loop

- Write a program that generates the following pattern.

Output

```
*  
++  
***  
++++  
*****
```

# Nested Loop

- Write a program that generates the following pattern.

```
int i, j;
for(i=1; i<=5; i++)
{
    for(j=1; j<=i; j++)
    {
        if (i % 2 == 0)
            printf("+");
        else
            printf("*");
    }
    printf("\n");
}
```

Output

```
*
++
***
++++
*****
```

# Nested Loop

- Write a program that generates the following pattern.

Output

```
      *  
     **  
    ***  
   ****  
  *****
```

# Nested Loop

- Write a program that generates the following pattern.

```
for (i=1; i<=5; i++)  
{  
    for (j=5; j>i; j--)  
        printf(" ");  
    for (j=1; j<=i; j++)  
        printf("*");  
    printf("\n");  
}
```

Output

```
      *  
     **  
    ***  
   ****  
  *****
```

# Home-works

- 1. Calculate the sum of the following series, where  $n$  is provided as user input.

$$1 + 2 + 3 + 4 + \cdots + n$$

- 2. Write a program that calculates the factorial of a positive integer  $n$  provided as user input.

- 3. Write a program that calculates  $a^x$ , where  $a$  and  $x$  are provided as user inputs.

- 4. Calculate the sum of the following series, where  $x$  and  $n$  is provided as user input.

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots + \frac{x^n}{n!}$$

# Home-works

Write programs that generate the following patterns. In each case, number of lines is the input

```
* * * * *
```

```
* * * *
```

```
* * *
```

```
* *
```

```
*
```

```
      *
```

```
    * * *
```

```
  * * * * *
```

```
* * * * * *
```

```
* * * * * * *
```

```
* * * * * * * *
```

```
  * * * * * *
```

```
    * * * *
```

```
      * * *
```

```
        *
```

```
* * * * * * * * *
```

```
* * * *      * * * *
```

```
* * *      * * *
```

```
* *      * *
```

```
*      *
```

```
* *      * *
```

```
* * *      * * *
```

```
* * * *      * * * *
```

```
* * * * * * * * *
```

```
      *
```

```
    *  *
```

```
  *      *
```

```
*      *
```

```
*      *
```

```
*      *
```

```
  *      *
```

```
    *  *
```

```
      *
```

```
*
```

```
* *
```

```
* * *
```

```
* * * *
```

```
* * * * *
```

```
* * * *
```

```
* * *
```

```
* *
```

```
*
```