

# C Programming Assignment / Week 6

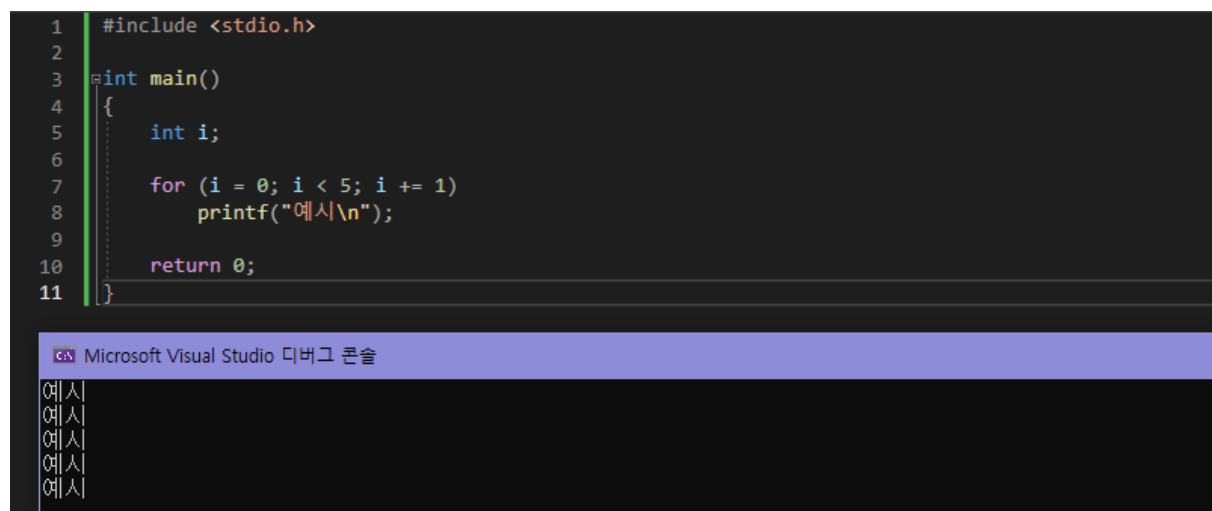
## 정리노트 #5

### 반복문

동일한 내용을 반복하거나, 일정한 규칙으로 반복하는 일을 수행할 때 필요하다.

- for 문

특정한 문장을 일정 횟수만큼 반복시킬 때 사용한다.



```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i;
6
7      for (i = 0; i < 5; i += 1)
8          printf("예시\n");
9
10     return 0;
11 }
```

Microsoft Visual Studio 디버그 콘솔

```
예시
예시
예시
예시
예시
```

초기화 (i=0): 변수의 초깃값을 지정하고, 반복문의 시작을 나타낸다

조건식(i<5): 변수값이 조건식에 맞으면 계속 반복하고 아니면 멈춤

명령문(printf("예시\n")); : 조건식이 참이면 명령문 수행

증감식(i += 1): 증감식에 따라 변수값을 증가시키거나 감소시킴

실행 순서는 초기화 > 조건식 > 명령문 > 증감식 이고, 종료 조건은 조건식이 거짓일 때 종료된다.

- 다중 for 문


반복문 안에 새로운 반복문이 들어가 있는 형태이다.

(사진 내 주석으로 설명)

```

1  #if 01
2
3  #include <stdio.h>
4
5  int main()
6  {
7      int i, j; // i와 j를 정수형 변수로
8      int sum = 0; // i와 j의 합을 저장하는 정수형 변수
9
10     for (i = 1; i <= 3; i++) { // 1. 초기화와 조건식을 실행하고 / 5. 증감식(i = i + 1)을 실행한다. 조건식에 부합하면 다시 2번으로 돌아가고 조건식에 맞지 않으면 6번으로 간다.
11         for (j = 1; j <= 3; j++) { // 2. 안에있는 두번째 for문의 초기화와 조건식을 실행한다. / 4. 증감식(j = j + 1)을 조건식에 맞지 않을때까지 실행하고 맞지 않으면 5번으로 간다.
12             sum = i + j;
13             printf("%d + %d = %d\n", i, j, sum); // 3. 중괄호 안에 묶여있는 두 명령문을 실행한다.
14         }
15     }
16
17     return 0; // 6. 다중 for문을 종료한다.
18 }
19 #endif

```



- while 문

횟수를 정확하게 알지는 못하지만 반복의 조건을 알고 있을 때 사용한다

```

53  #if 01
54
55  #include <stdio.h>
56
57  int main()
58  {
59      int sum = 0, number = 1;
60      printf("1부터 10까지의 합: ");
61
62      while (number <= 10)
63      {
64          sum += number;
65          number++;
66      }
67      printf("%d", sum);
68  }
69  #endif
70

```



- do~while

일단 do 안에 있는 명령문을 수행한 후, 나중에 while 문의 조건식을 비교한다.

조건식을 검사하는 while(조건식)은 c 언어 문장 처리되므로 세미콜론을 반드시 써야한다.

기타 제어문

- goto 문

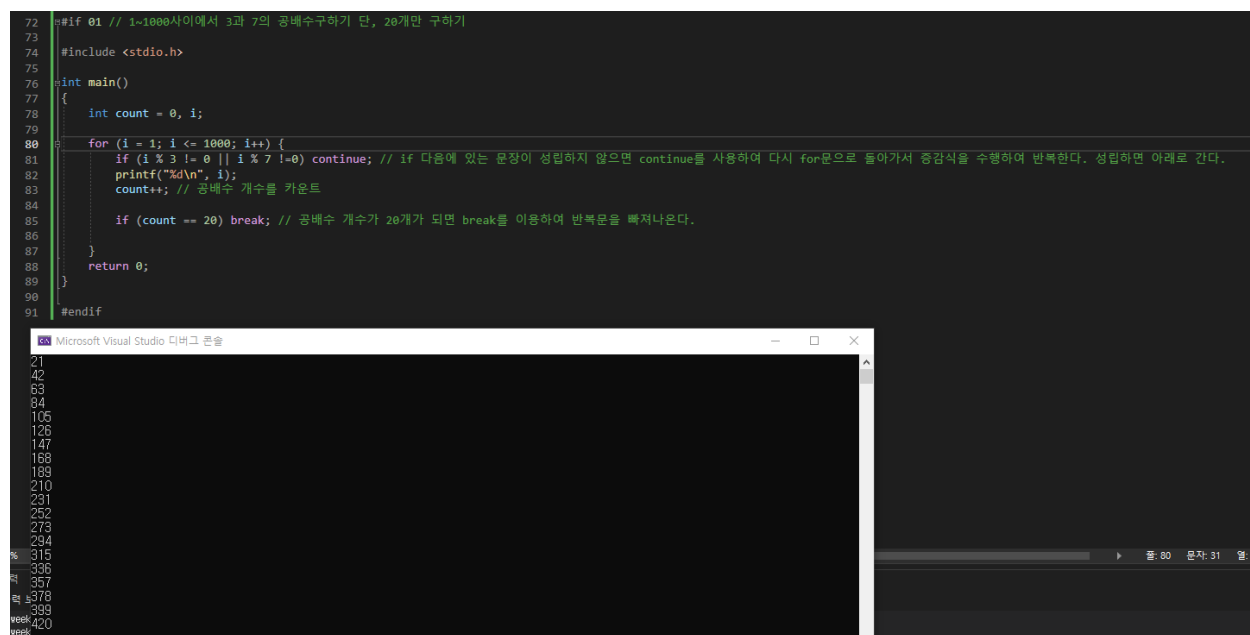
명령문의 실행 순서를 프로그래머가 임의로 변경하려고 할 때 사용

goto 문을 남발하면 에러가 생기기 때문에 잘 사용하지는 않는다.

- break 문, continue 문

break 문은 반복문을 빠져나오거나, switch~case 문에서 정상적인 흐름을 변경하기 위하여 사용한다.

continue 문은 break 문과 대립되는 제어문으로 반복문을 빠져나오지 않고, 해당 반복문 o 의 처음으로 프로그램의 흐름을 이동시킨다.



```
72 //if 01 // 1~1000사이에서 3과 7의 공배수구하기 단, 20개만 구하기
73
74 #include <stdio.h>
75
76 int main()
77 {
78     int count = 0, i;
79
80     for (i = 1; i <= 1000; i++) {
81         if (i % 3 != 0 || i % 7 != 0) continue; // if 다음에 있는 문장이 성립하지 않으면 continue를 사용하여 다시 for문으로 돌아가서 중감식을 수행하여 반복한다. 성립하면 아래로 간다.
82         printf("%d\n", i);
83         count++; // 공배수 개수를 카운트
84
85         if (count == 20) break; // 공배수 개수가 20개가 되면 break를 이용하여 반복문을 빠져나온다.
86     }
87     return 0;
88 }
89
90 #endif
```

수업시간에 소수 구하기도 교수님이 한번 만들어 보라 하셨는데, 바로 만들어보려 했으나, 어떻게 해야 할지 막막해서 정리노트를 작성하며 복습하며 만들어봤다.

```
112 #endif 01 // 1부터 100까지의 소수
113
114 #include <stdio.h>
115
116 int main()
117 {
118     int i, j; // 변수 2개 지정
119     for (i = 2; i <= 100; i++) { // 소수인지 아닌지 판별될 숫자, 1~100까지의 범위 지정
120         for (j = 2; j <= i; j++) { // 소수를 판별하기 위한 변수
121             if (i % j == 0) { // (판별될 숫자) 나누기 (판별하기 위한 변수) 의 나머지가 0이면 소수가 되기 위한 1번째 조건 완성
122                 break;
123             }
124         }
125         if ( i == j ){ // 나머지가 0인 것 외에도, 소수는 1또는 자기 자신으로 나누었을 때의 나머지가 0이 되어야 하므로 두 숫자가 같으면 소수가 되기 위한 2번째 조건
126             printf("%d ", i); // 판별된 소수만 출력
127         }
128     }
129 }
130
131 #endif
132
133
134 Microsoft Visual Studio 디버그 콘솔
135
136 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

124 번째 줄까지는 무난하게 갔으나, 거기까지 하고 컴파일을 했을 때, 소수가 아닌 수도 나오고 같은 숫자가 여러번 나오고 이상한 결과가 출력되었다. 다시 한번 생각을 해 보니, 소수가 되기 위한 조건에는 나머지가 0 인것 뿐 아니라 1 또는 자기 자신만 되어야 소수가 되는 것을 깨닫고 나서, if 문을 하나를 더 추가하여 정상 출력 되었다.