

4 장 선택 반복 제어

□ 개념 확인 학습

1. C프로그램에서 if는 어떻게 사용하나요?
2. C프로그램에서 switch는 어떻게 사용하나요?
3. C프로그램에서 switch를 사용할 경우 case에 break가 존재할 때와 아닐 때가 어떻게 다른가요?
4. 반복 제어 do ~ while과 while의 차이는 무엇인가요?
5. 반복 제어 for의 요소들 초기식, 조건식, 증감식, 명령문의 수행 순서는 어떻게 되나요?
6. 반복 제어 for의 블록 명령문이 모두 끝나는 시점은 언제인가요?
7. 반복 제어 문장에서 break는 언제 사용하나요?

8. 반복 제어 문장에서 continue는 언제 사용하나요?
9. while, do~while, for 반복의 내부 블록에서 continue를 만났을 때의 수행 과정은 어떻게 되나요?
10. 소문자를 대문자로, 대문자를 소문자로 바꾸고 싶을 때는 어떤 함수를 사용하나요?

☐ 적용 확인 학습

1. 다음 서술 내용이 바르면 O, 그렇지 않으면 X를 표시 하세요.
- A. 선택 제어 if와 else는 반드시 쌍으로 존재해야 한다. ()
- B. if, else if를 사용하면 여러 가지 조건을 검사할 수 있다. ()
- C. switch(정수표현식)에서 정수표현식에는 int, short, char를 반환하는 식이 포함 될 수 있다. ()
- D. switch의 case에는 1, 0과 같이 여러 개의 조건을 검사할 수 있다. ()
- E. break는 반복의 종료에만 사용된다. ()

- F. break는 반복의 종료에 사용되며, 중첩된 모든 반복을 탈출한다. ()
- G. do ~ while 반복은 조건을 먼저 평가하여 반복의 여부를 결정한다. ()
- H. while 반복은 조건을 먼저 평가하여 반복의 여부를 결정한다. ()
- I. for 반복은 조건의 평가에서 종료하게 된다. ()
- J. for(; ;)처럼 for 반복의 괄호 내부의 모든 항목은 생략 가능하다. ()
- K. for반복의 괄호 내부에서 2개의 세미콜론은 반드시 필요하다. ()
- L. do { ... } while(1); while(1) { ... }; for(1) { ... };은 무한 반복을 의미한다.
()
- M. for 반복의 내부에서 continue를 만나면 조건을 평가하는 곳으로 이동한다. ()
- N. switch의 case에 값으로 문자열을 사용할 수 있다. ()

2. 다음에 서술된 문장에 어울리는 프로그램 코드를 완성해 보세요.

- A. 문자형 변수 ch가 ‘a’ 이면 “apple” 을 출력하고, 그렇지 않으면 “fruit” 을 출력한다. (switch 사용)
- B. 정수형 변수 나이(age)가 12세 미만이면 “놀아 줄까요?”, 12세 이상 20 이하면 “고기 사줄까요?”, 그렇지 않으면 “용돈 드릴까요?” 를 출력

한다. (if 사용)

C. 'a' 부터 'z' 까지 출력한다. (do~while, while, for 각각 사용)

D. 1부터 10까지 출력한다. (do~while, while, for 각각 사용)

□ 응용 프로그래밍

>> 응용 프로그래밍은 다음과 같은 구조로 작성합니다. 번호가 늘어 갈 경우 번호에 해당하는 함수를 만들어 사용합니다.

```
#include <stdio.h>
```

```
void ex_02(void) {
```

```
    //응용 프로그래밍 2번은 ex_02() 내부에 작성합니다
```

```
}
```

```
void ex_03(void) {
```

```
    //printf("응용 프로그래밍 3번은 ex_03() 내부에 작성합니다.
```

```
}
```

```
int main() {
```

```
    ex_02();
```

```
    ex_03();
```

```
    return 0;
```

```
}
```

1. 입력받은 정수가 음수인지 양수인지 판별하세요. (if 사용)

정수를 입력하세요 : 5
양수입니다.

정수를 입력하세요 : -5
음수입니다.

2. 입력받은 정수를 2로 나눈 몫이 10보다 크면 “OK” 그렇지 않으면 “NOK” 를 출력하는 프로그램을 작성하세요. (switch 사용)

정수를 입력하세요 : 22
OK

3. 입력받은 숫자들을 모두 더하는 프로그램을 작성하세요. 단, 입력받은 수가 -100 이하 또는 100 이상이면 반복을 종료하고 입력받은 숫자들의 합을 출력하세요.

숫자를 입력하세요 : 22.22
숫자를 입력하세요 : 55
숫자를 입력하세요 : 11.1
숫자를 입력하세요 : 1000
-100 이상 100 이하 숫자들의 합 = 88.32

4. 문자를 입력받아 화면에 출력하는 작업을 반복하는 프로그램을 작성하세요. 단, 입력 받은 문자가 ‘q’ 또는 ‘Q’ 이면 반복을 종료합니다.

문자를 입력하세요.
a
p
p
l
e
q

5. 입력받은 정수를 2진수로 변환하여 출력하는 프로그램을 작성하세요.

정수를 입력하세요 : 64
2진수는 = 1000000

정수를 입력하세요 : 15
2진수는 = 1111

6. 양의 정수를 입력받아 그 정수를 2진수로 만들 때 1이 나타나는 횟수를 출력하세요. (양의 정수가 아닌 수를 입력받으면 양의 정수 다시 입력, 나누기와 나머지 연산자 사용 금지)

정수를 입력하세요 : 64
2진수로 만들 때 1이 나타나는 횟수 = 1

정수를 입력하세요 : 15
2진수로 만들 때 1이 나타나는 횟수 = 4

7. 10부터 입력받은 정수까지의 수에서 2, 3, 5, 7의 배수를 제외한 수를 한 행에 10개씩 출력하는 프로그램을 작성하세요.

정수를 입력하세요 : 9
정수를 입력하세요 : 5
정수를 입력하세요 : 55
11 13 17 19 23 29 31 37 41 43
47 53

8. 입력받은 알파벳 문자부터 'Z' 까지 한 행에 5개씩 출력하세요. 단, 입력받은 알파벳 문자는 대소문자 여부에 관계없이 대문자로 사용합니다.

알파벳 입력 : @
알파벳 입력 : 5
알파벳 입력 : r
R S T U V
W X Y Z

9. 실행화면을 참고하여 프로그램을 작성하세요.

```
k이하의 알파벳입력(q는 종료, k이상은 다시 입력) : c
abc
k이하의 알파벳입력(q는 종료, k이상은 다시 입력) : l

k이하의 알파벳입력(q는 종료, k이상은 다시 입력) : h
abcdefgh
k이하의 알파벳입력(q는 종료, k이상은 다시 입력) : q
q입력으로 종료
```

10. 실행화면을 참고하여 프로그램을 작성하세요.

```
문자입력(알파벳과 숫자 이외의 문자 입력 시 종료) : c
소문자입니다
문자입력(알파벳과 숫자 이외의 문자 입력 시 종료) : 2
숫자입니다. 지금까지 숫자들의 합 = 2
문자입력(알파벳과 숫자 이외의 문자 입력 시 종료) : 5
숫자입니다. 지금까지 숫자들의 합 = 7
문자입력(알파벳과 숫자 이외의 문자 입력 시 종료) : B
대문자입니다
문자입력(알파벳과 숫자 이외의 문자 입력 시 종료) : #
종료합니다.
```

11. 이 문제의 힌트를 실행화면의 맨 마지막 줄 ‘765432101234567’을 설명합니다. 숫자들이 수직 선상에 놓여 있고, 0의 왼쪽은 음수, 오른쪽은 양수라고 생각해 보세요. 그렇게 생각하면 음수부터 양수까지 출력을 한 것입니다. 이제 남은 문제는 음수의 부호를 어떻게 빼는가 하는 것입니다. 음수의 부호는 절대값을 알려주는 `abs()` 함수를 사용하여 없앨 수 있습니다. `abs()`는 `stdlib.h`를 필요로 합니다.

```
0
101
21012
3210123
432101234
54321012345
6543210123456
```

이제 실행화면을 참고하여 프로그램을 작성하세요.

765432101234567

12. 실행화면을 참고하여 사용자가 입력한 개수만큼 알파벳을 입력받아 알파벳 순서에서 ‘m’ 과 가장 가까운 문자를 출력하는 프로그램을 작성하세요. 단, 입력받은 알파벳 문자는 대소문자 여부에 관계없이 소문자로 사용합니다.

반복횟수입력: 4

문자입력: t

문자입력: h

문자입력: b

문자입력: k

m과 가장 가까운 문자는 = k