

2023-02학기 C프로그래밍 중간고사 해설지

{문제 1 - 체크리스트}

문제 번호	제출한 파일명에 동그라미	실행(O) 부분실행(▲) 실행안됨(X)
문제2. (5점)	02.c	O
문제3. (5점)	03.c	O
문제4. (5점)	04.c	O
문제5. (5점)	05.c	O
문제6. (5점)	06.c	O
문제7. (5점)	07.c	O
문제8. (5점)	08.c	O
문제9. (5점)	09.c	O

{문제 2 - 배열}

일기장 복사 프로그램

- 키보드를 통한 간단한 일기를 쓰자
- 100byte 이하, 배열필요
 - Char diary[MAX] 필요
 - 일기를 배열에 저장
- 배열에 저장된 일기를 모니터 출력
 - 단, 공백도 일기에 저장되어야 함
- 일기장 복사본을 만들자
 - strcpy()를 사용
 - char copy_diary[MAX]의 복사본 배열 출력
 - 복사본 일기를 모니터에 출력

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <limits.h>      // 최대 최소 CHAR_MAX
4  #include <string.h>      // strcpy() 함수가 선언된 헤더 파일
5
6  int main(void) {
7      char diary[CHAR_MAX];      // 최대 크기로 char형 배열을 선언
8      char copy_diary[CHAR_MAX]; // 최대 크기로 char형 배열을 선언
9
10     printf("[입력]\n");
11     scanf("%[^\n]s", diary);    // 서식 지정자 %[^\n]s 필수
12     printf("[작성된 일기]\n%s\n", diary);
13     strcpy(copy_diary, diary);  // diary 문자열을 copy_diary로 복사
14     printf("[복사된 일기]\n%s\n", copy_diary);
15
16     return 0;
17 }
```

{문제 3 - 배열 & 함수}

원의 둘레, 원의 넓이 프로그램

- scanf() 를 통해 반지름 입력
- 원의 둘레 함수, 원의 넓이 함수 구현
 - 데이터는 모두 실수형(double)
 - 소수점 둘째 자리까지만 출력
 - 반드시 함수로 구현되어야 함

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  double calculate_cm2_circle(double);
5  double calculate_cm_circle(double);
6  int main(void) {
7      double n;
8      printf("반지름 입력(cm): ");
9      scanf("%lf", &n);
10     printf("원의 넓이 %.2lfPI(cm2)\n", calculate_cm2_circle(n));
11     printf("원의 둘레 %.2lfPI(cm)\n", calculate_cm_circle(n));
12
13     return 0;
14 }
15
16 double calculate_cm2_circle(double n) {
17     double result = 0;
18     result = n * n;
19
20     return result;
21 }
22
23 double calculate_cm_circle(double n) {
24     double result = 0;
25     result = 2 * n;
26
27     return result;
28 }
29 }
```

{문제 4 - 배열 & 함수 & 암호화}

암호화 프로그램

- 사용자 비밀번호 3개 키보드 입력
 - `int get_num()` 사용, 221페이지
 - `int user[MAX]`에 저장, 반복문
 - `user[0]`, `user[1]`, `user[2]`에 저장
- 비밀번호 암호화 함수 구현
 - `int encrypt(int);`
 - 반드시 함수로 구현되어야 함
 - `KEY = 2023`
 - XOR (비트연산자, 진리표 참조)
- 암호화된 비밀번호 저장 및 출력
 - `int encryption[MAX]`에 저장, 반복문

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  #define KEY 2023    // 암호화에 사용할 키 값
5  #define MAX 3      // 사용자 수를 3으로 정의
6
7  int get_num();
8  int encrypt(int);
9
10 int main(void) {
11     int user[MAX];
12     int encrypted[MAX];
13     int decrypted[MAX];
14
15     for (int i = 0; i < MAX; i++) {
16         user[i] = get_num();
17         printf("%d번째 사용자 비밀번호 입력문: %d\n", i, user[i]);
18         encrypted[i] = encrypt(user[i]);
19         printf("%d번째 사용자 비밀번호 암호문: %d\n", i, encrypted[i]);
20         printf("-----\n");
21     }
22 }
23
24 int get_num() {
25     int num;
26     scanf("%d", &num);
27     return num;
28 }
29
30 int encrypt(int num) {
31     return num ^ KEY;
32 }
```

{문제 5 - 배열 & 함수 & 복호화}

복호화 프로그램, 문제 4와 연계

- 문제 4에서 사용된 암호문 배열 이용
 - `int encryption[MAX]` 재사용
- 암호문 to 복호화 함수 구현
 - `int decrypt(int);`
 - ※ 반드시 함수로 구현되어야 함
 - `KEY = 2023`
 - XOR (비트연산자, 진리표 참조)
- 복호화된 비밀번호 바로 출력

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  #define KEY 2023    // 암호화에 사용할 키 값
5  #define MAX 3      // 사용자 수를 3으로 정의
6
7  int get_num();
8  int encrypt(int);
9  int decrypt(int);
10
11 int main(void) {
12     int user[MAX];
13     int encrypted[MAX];
14     int decrypted[MAX];
15
16     for (int i = 0; i < MAX; i++) {
17         user[i] = get_num();
18         printf("%d번째 사용자 비밀번호 입력문: %d\n", i, user[i]);
19         encrypted[i] = encrypt(user[i]);
20         printf("%d번째 사용자 비밀번호 암호문: %d\n", i, encrypted[i]);
21         decrypted[i] = decrypt(encrypted[i]);
22         printf("%d번째 사용자 비밀번호 복호문: %d\n", i, decrypted[i]);
23         printf("-----\n");
24     }
25 }
26
27 int get_num() {
28     int num;
29     scanf("%d", &num);
30     return num;
31 }
32
33 int encrypt(int num) {
34     return num ^ KEY;
35 }
36
37 int decrypt(int num) {
38     return num ^ KEY;
39 }
```

{문제 6 - 배열 & 수열 & 함수 재사용}

수열 입력 & 수열 합 프로그램

- 수열 홀수 5개 입력 -> 배열 저장
- 수열 입력 함수, 수열 합 함수 구현
 - `double get_num()` 사용, 221.p
 - `void print_line()` 사용, 225.p
 - `double sum()` 개발
 - call by value

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  double get_num(void);
5  void print_line(void);
6  double sum(double arr[], int size);
7
8  int main(void) {
9      double num[5];
10
11     printf("수열 5개 입력 (예: 1 3 5 7 9): ");
12     for (int i = 0; i < 5; i++) {
13         num[i] = get_num();
14     }
15
16     print_line();
17     printf("입력된 수열 확인: ");
18     for (int i = 0; i < 5; i++) {
19         printf("%.2lf ", num[i]);
20     }
21     printf("\n");
22     print_line();
23
24     double result = sum(num, 5);
25     printf("수열의 합: %.2lf\n", result);
26
27     print_line();
28     return 0;
29 }
30
31 // double arr[] **
32 double sum(double arr[], int size) {
33     double total = 0;
34     for (int i = 0; i < size; i++) {
35         total += arr[i];
36     }
37     return total;
38 }
39
40 void print_line(void) {
41     printf("-----\n");
42 }
43
44 double get_num(void) {
45     double num;
46     scanf("%lf", &num);
47     return num;
48 }
```

{문제 7 - 반복문 & 함수}

대소문자 반복 변경 프로그램

- 문자 하나를 입력받아, 대문자/소문자를 판별하는 `char check(char ch)` 함수 구현
 - 대문자이면 소문자로 출력
 - 소문자이면 대문자로 출력
 - 무한루프로 계속 진행
 - 숫자를 입력하면 무한루프 탈출

```
1  #include <stdio.h>
2
3  char check(char ch) {
4      if (ch >= 'A' && ch <= 'Z') {
5          return ch + 32; // ASCII에서 대문자와 소문자의 차이는 32
6      } else if (ch >= 'a' && ch <= 'z') {
7          return ch - 32;
8      } else {
9          return ch;
10     }
11 }
12
13 int main() {
14     char input;
15
16     while (1) {
17         printf("문자 입력: ");
18         scanf(" %c", &input);
19
20         // 교수님께 질문하기
21         if (input >= '0' && input <= '9') {
22             printf("반복 문자 입력을 종료합니다.\n");
23             break;
24         }
25
26         char result = check(input);
27         printf("변경 전: %c 변경 후: %c\n", input, result);
28         printf("-----\n");
29     }
30
31     return 0;
32 }
```

{문제 8 - 반복문 & 나머지연산 & 함수}

선택적 구구단 프로그램

- void gugudan(int) 함수 구현
 - 1 ~ 9 사이의 정수 입력
 - 정수가 홀수면, '홀수 단'만 출력
 - 정수가 짝수면, '다시 입력' 출력
 - 무한 루프를 통해 반복해야 함
 - 입력된 수 0 이면, 무한루프 탈출 하여 프로그램 종료
 - ※ 반드시 함수로 구현되어야 함

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  void gugudan(int num) {
5      while (1) {
6          if (num == 0) {
7              printf("숫자 입력을 종료합니다.\n");
8              return;
9          }
10
11         if (num < 1 || num > 9) {
12             printf("-----\n");
13             printf("1부터 9 사이의 숫자를 입력하세요.\n");
14             printf("-----\n");
15         } else if (num % 2 == 0) {
16             printf("-----\n");
17             printf("다시 입력 하세요.\n");
18             printf("-----\n");
19         } else {
20             printf("-----\n");
21             for (int i = 1; i <= 9; i++) {
22                 printf("%d * %d = %d\n", num, i, num * i);
23             }
24             printf("-----\n");
25         }
26
27         printf("숫자 입력: ");
28         scanf("%d", &num);
29     }
30 }
31
32 int main() {
33     int input;
34     printf("숫자 입력: ");
35     scanf("%d", &input);
36
37     gugudan(input);
38
39     return 0;
40 }
```

{문제 9 - 무한반복문 & switch ~ case}

무한 반복 숫자비교 프로그램

- `int checkInt(int a, int b)` 함수 정의
 - 두 숫자를 비교하는 함수
 - 입력값: 사용자가 정의한 2개 숫자
 - 출력값: $a > b$ 일 때 1,
 $a < b$ 일 때 2,
 $a == b$ 일 때 3 리턴
- ※ 무한반복문으로 숫자 입력해야 함

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3
4  int checkInt(int a, int b) {
5      if (a > b) {
6          return 1;
7      } else if (a < b) {
8          return 2;
9      } else {
10         return 3;
11     }
12 }
13
14 int main() {
15     int num1, num2;
16     char input[20];
17
18     while (1) {
19         printf("2개 숫자 입력(예: 2 5): ");
20         fgets(input, sizeof(input), stdin);
21
22         if (sscanf(input, "%d %d", &num1, &num2) == 2) {
23             int result = checkInt(num1, num2);
24
25             switch (result) {
26                 case 1:
27                     printf("큰 값: %d, 작은 값: %d\n", num1, num2);
28                     printf("-----\n");
29                     break;
30                 case 2:
31                     printf("큰 값: %d, 작은 값: %d\n", num2, num1);
32                     printf("-----\n");
33                     break;
34                 case 3:
35                     printf("두 수가 같다.\n");
36                     printf("-----\n");
37                     break;
38             }
39             } else {
40                 printf("숫자를 입력하세요. 프로그램을 종료합니다.\n");
41                 break;
42             }
43     }
44
45     return 0;
46 }
```