

문항 선택

i

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

한 문제씩 검토

검토 완료

강의실 홈

강의정보

강의계획서

성적/출석관리

학습진도현황

오프라인출석부

성적부

수강생 알림

쪽지 보내기

기타 관리

학습활동

🏠

C프로그래밍

[MidTerm-A:100]

| | |
|-------|----------------------|
| 시작 일시 | 2018-10-29, 18:03:53 |
| 진행 상황 | 종료됨 |
| 완료 일시 | 2018-10-29, 18:54:11 |
| 소요시간 | 50 분 18 초 |
| 성적 | 최고 100.00점 중 90,50점 |

정보

The total points of the mid-term exam is 200.

The following table shows points assigned to each question.

Midterm-A (Short Answer Questions - 100 points, Exam Time - 55 minutes : 18:00 ~ 18:55)

| Q # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| pts | 4 | 5 | 4 | 5 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 6 | 5 | 5 | 6 | 5 | 5 | 5 | 5 | 4 |

Midterm-B (Essay(Programming) Questions - 100 points, Exam Time - 80 minutes : 19:00 ~ 20:20)

| Q # | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|----|----|----|----|----|----|
| pts | 10 | 15 | 15 | 20 | 20 | 20 |

문제 1

정답

총 4.00 점에서 4.00 점 할당

What is the user demand calculation (operation) that can be translated into the following sequence of instructions ? 2 ✓

| Sequence of Instructions | Description about Instructions |
|---|--|
| ① MUL A B R1 ② MUL R1 2 R2 ③ ADD R2 B C | A, B, C, R1, and R2 : storage ADD, MUL : opcode ADD X Y Z : Z←X+Y; MUL X Y Z : Z←X*Y |

1. $C \leftarrow A \times A + A \times B + A$

2. $C \leftarrow A \times B + A \times B + B$

3. $C \leftarrow A \times B + B + A$

4. $C \leftarrow A \times A + A + B$

문제 2

부분적으로 맞음

총 5.00 점에서 4.00 점 할당

For each of the followings, determine whether it is a correct description for the C language or not.

- It translates the entire source code into binary machine code and bundles it into a binary executable. O ✓
- It translates part of the source code (usually one sentence) into binary machine code and runs the results immediately. X ✓
- The tool used to translate a C source code into binary machine code is called an Interpreter. X ✓
- If we have a binary executable file generated from a C source code, we do not need the source code for execution. X ✗
- As the C language is good in portability, we don't have to recompile and regenerate a binary executable even if the target machines are different in their CPU/ISA. X ✓

문제 3

정답

총 4.00 점에서
4.00 점 할당

For [1]~[7] shown in the source code below, select a word that best describes it.
Words you can choose : Block, Comment, Configuration, Declaration, Function Definition, Function Prototype, Instruction, Operation, Preprocessor Directive, Statement

- [1] :

Comment

 ✓
- [2] :

Preprocessor Directive

 ✓
- [3] :

Function Prototype

 ✓
- [4] :

Function Definition

 ✓
- [5] :

Block

 ✓
- [6] :

Declaration

 ✓
- [7] :

Statement

 ✓

```
/* This is a Sample Code */  
  
#include <stdio.h>  
void greeting(void);  
  
int main(void) {  
    int a;  
    int b=3;  
    a=3;  
    b=a*b;  
    {  
        int c;  
        c=a-b;  
        b=c*3;  
    }  
    greeting();  
    return 0;  
}  
  
void greeting(void) {  
    printf("Hello, World");  
}
```

Diagram annotations:

- [1] points to the first comment line.
- [2] points to the preprocessor directive line.
- [3] points to the function prototype line.
- [4] points to the main function definition block (both header and body).
- [5] points to the inner block within main.
- [6] points to the variable declarations within main.
- [7] points to the first statement within main.

문제 4
부분적으로 맞음
총 5.00 점에서
4.50 점 할당

For the source code below, determine whether it raises a syntax error or not.
Select "E" if it raise a syntax error. Select "X", if it is not an error.

```
#include <stdio.h>
int main(void)
{
    int _id;           // X ✓
    int not#me;        // E ✓
    int 010_south;     // E ✓
    int so_am_i;       // X ✓
    int star*it;       // E ✓
    int -plus;         // E ✓
    int me_to-2;       // E ✓
    int long;          // E ✓
    int break;         // E ✓
    int scanf;         // E ✗

    return 0;
}
```

문제 5

정답

총 4.00 점에서
4.00 점 할당

Write down the output of each printf() function from the below program.

The FSF "%.nlf" means to output the value of a double variable with n digits after ".". For example, "%.1lf" specifies the number of digits after "." is 1. If the value to be printed is 1.200000, the output will be 1.2.

| | |
|---------------|-------|
| // 1 printf() | 1 ✓ |
| // 2 printf() | 2 ✓ |
| // 3 printf() | 3 ✓ |
| // 4 printf() | 0 ✓ |
| // 5 printf() | 1.0 ✓ |
| // 6 printf() | 0.6 ✓ |
| // 7 printf() | 0.0 ✓ |

```
#include <stdio.h>
int main(void)
{
    int    na=5,  nb=3,  nx;
    double da=4.0, db=3.2, dx;

    nx = na/nb;
    printf("%d\n", nx);    // 1

    nx = na*nb;
    printf("%d\n", nx);    // 2

    nx = db;
    printf("%d\n", nx);    // 3

    nx = db/da;
    printf("%d\n", nx);    // 4

    dx = na/nb;
    printf("%.11f\n", dx); // 5

    dx = (double)nb/na;
    printf("%.11f\n", dx); // 6

    dx = (int)db/na;
    printf("%.11f\n", dx); // 7

    return 0;
}
```

문제 6

정답

총 4.00 점에서
4.00 점 할당

What are the types of conversion in the following example ?

- A : Implicit ✓ Extended ✓ Conversion
- B : Implicit ✓ Narrowed ✓ Conversion

```
#include <stdio.h>
int main(void)
{
    int a = 1, b=2;
    float c=2.0, d=2.0;

    c = a / 2;    //A
    b = c / d;    //B
    return 0;
}
```

문제 7

정답

총 5.00 점에서
5.00 점 할당

It is known that the normal range of normal pulse rate is between 60 and 90. That is, $60 \leq \text{normal pulse rate} \leq 90$.

We want to check if the input pulse rate belongs to the normal range. Choose all of the conditional expressions that are valid for [1].

```
#include <stdio.h>
int main(void)
{
    int nPulseRate;

    printf(" Input pulse rate: ");
    scanf("%d", &nPulseRate);

    if ([1])
        printf("\n%d is in a normal range.\n", nPulseRate);

    return 0;
}
```

하나 이상을 선택하세요.

- ☐ 1. $60 \leq \text{nPulseRate} \leq 90$
- ☐ 2. $!(60 \leq \text{nPulseRate} \leq 90)$
- ☐ 3. $\text{nPulseRate} \geq 60 \ \& \ \text{nPulseRate} \leq 90$
- ☒ 4. $\text{nPulseRate} \geq 60 \ \&\& \ \text{nPulseRate} \leq 90$ ✓
- ☐ 5. $\text{nPulseRate} < 60 \ || \ \text{nPulseRate} > 90$
- ☒ 6. $!(\text{nPulseRate} < 60 \ || \ \text{nPulseRate} > 90)$ ✓

답이 맞습니다.

정답 : $\text{nPulseRate} \geq 60 \ \&\& \ \text{nPulseRate} \leq 90, !(\text{nPulseRate} < 60 \ || \ \text{nPulseRate} > 90)$

문제 8

정답

총 5.00 점에서
5.00 점 할당

In the following code, we want to determine whether the 2nd last digit of the int variable a is even or odd. For example if a is 374, the 2nd digit number, 7, is determined into odd. Complete the code by filling the blank:

You can safely assume that the integer variable a has more than three digits. That is, $a \geq 100$.

DO NOT include inessential white spaces.

```
#include <stdio.h>
int main(void)
{
    int a;

    printf("Input an integer which is larger than 100: ");
    scanf("%d", &a);

    printf("The 2nd last number of %d", a);
    if ( a/10%10%2 )
        printf(" is odd\n");
    else
        printf(" is even\n");

    return 0;
}
```

댓글:

문제 9

틀림

총 5.00 점에서
0.00 점 할당

What is the output of the following program ?

Execution Result

w

✖

```
#include <stdio.h>

int main(void)
{
    char ch='s';
    while (ch<'w')
        printf("%c",ch++);
    printf("%c\n",ch);
    return 0;
}
```

문제 10

정답

총 4.00 점에서
4.00 점 할당

Compare the following 2 programs.
If executed, **how many times** does each program outputs the string "found" ?

Program 1 : ✔, Program 2 : ✔

| Program 1 | Program 2 |
|--|---|
| <pre>#include <stdio.h> int main(void) { int i; // index int a[10]={3,2,8,0,2,3,5,6,4,2}; for(i=9; i>=0; i--) { if (a[i]%2) { puts("found"); } } return 0; }</pre> | <pre>#include <stdio.h> int main(void) { int i; // index int a[10]={3,2,8,0,2,3,5,6,4,2}; for(i=9; i>=0; i--) { if (a[i]%2) { puts("found"); break; } } return 0; }</pre> |

문제 11

정답

총 4.00 점에서
4.00 점 할당

Write the output of the following program. Fill in each blank with an appropriate number in printf().

4



2



1



3



```
#include <stdio.h>
double square(double x) {
    printf("1 ");
    return x * x;
}

double cube(double x) {
    printf("2 ");
    return x * square(x);
}

double pi(double x) {
    printf("3 ");
    return 3.14 * x;
}

double sphere(double r) {
    printf("4 ");
    return 4.0 / 3.0 * pi(cube(r));
}

int main(void) {
    double radius = 1.5;
    double volume;

    volume = sphere(radius);
    return 0;
}
```

문제 12

부분적으로 맞음

총 6.00 점에서
5.00 점 할당

In terms of Scope, variables can be classified into two types.

- In the case of a variable declared in a function area, it is valid within the function and is called a variable.
- If a variable is declared outside the function, it is valid in the entire function of the source file and is called a variable.

Write the output of the following program.

Execution Result

s1= , s2=

```
#include <stdio.h>
int gSum=0;
int sum1(int n)
{
    gSum = gSum + n;
    return gSum;
}
int sum2(int n)
{
    int gSum = 0;
    gSum = gSum + n;
    return gSum;
}
int main(void)
{
    int n1=1,n2=10,n3=100;
    int s1, s2;

    s1 = sum1(n1); s1 = sum1(n2); s1 = sum1(n3);
    s2 = sum2(n1); s2 = sum2(n2); s2 = sum2(n3);

    printf("s1=%d,s2=%d\n",s1,s2);
    return 0;
}
```

문제 13

정답

총 5.00 점에서
5.00 점 할당

For the following user input (which of each is stored to a variable **n**), write the output value of the last printf() function in main().

| User Input | The output of the last printf() |
|------------|--|
| 1 | hanoi_tower() called <input type="text" value="1"/> ✓ times |
| 2 | hanoi_tower() called <input type="text" value="3"/> ✓ times |
| 3 | hanoi_tower() called <input type="text" value="7"/> ✓ times |
| 4 | hanoi_tower() called <input type="text" value="15"/> ✓ times |
| 5 | hanoi_tower() called <input type="text" value="31"/> ✓ times |

```
#include <stdio.h>
int n_hanoi_called;
void hanoi_tower(int n, int ox, int tx, int mx) {
    n_hanoi_called++;
    if (n < 1) {
        printf("Error: n >= 1\n");
    } else if (n == 1) {
        printf("%d -> %d\n", ox, tx);
    } else {
        hanoi_tower(n-1, ox, mx, tx);
        printf("%d -> %d\n", ox, tx);
        hanoi_tower(n-1, mx, tx, ox);
    }
}
int main(void) {
    int n;
    n_hanoi_called = 0;
    printf("Enter the height of the tower\n");
    scanf("%d",&n);
    hanoi_tower(n, 1, 3, 2);
    printf("hanoi_tower() called %d times\n",
           n_hanoi_called);
    return 0;
}
```


문제 14

부분적으로 맞음

총 5.00 점에서
4.00 점 할당

What is the output of the following program ? Fill in the blanks.

Execution Result

| | |
|---|---|
| 1 | ✓ |
| 0 | ✗ |
| 0 | ✓ |
| 4 | ✓ |
| 1 | ✓ |

```
#include <stdio.h>

int main(void)
{
    int a=5, b=2, c=4, d=5;

    printf("%d\n", a == 5);
    printf("%d\n", b < d < c);
    printf("%d\n", b % c * a && a % d * b);
    printf("%d\n", a % b * c);
    printf("%d\n", d % b * c > 5 || c % b * d < 7);

    return 0;
}
```

문제 15

정답

총 6.00 점에서
6.00 점 할당We want to store a value, **4312**, in a computer system.

To be stored in a computer, all information should be encoded into binary values.

We have two possible choices in this case.

- 1) Encode each digit as an **ASCII** character.
 - 4312 has 4 digits and each digit require 1 byte. So we need **memory space by 4 bytes**.
 - Let the following figure represents the 4 bytes memory space for 4312.
Fill in the blanks with suitable binary values in **Hexadecimal format**.

| | | | | | | | | | |
|------|----|----|---|----|----|---|----|----|---|
| 0x34 | 0x | 33 | ✓ | 0x | 31 | ✓ | 0x | 32 | ✓ |
|------|----|----|---|----|----|---|----|----|---|

- 2) Encode 4312 as a **signed short** integer with **memory space by 2 bytes**.
 - Let the following figure represents the 2 bytes memory space for the value.
 - Fill in the blanks with appropriate binary values in **Hexadecimal format**.

| | | | |
|------|----|----|---|
| 0x10 | 0x | D8 | ✓ |
|------|----|----|---|

문제 16

정답

총 5.00 점에서
5.00 점 할당

For each of the following programs, write its output.
Note that we do not align the indentation on purpose.

| | | |
|---|---|--|
| <div>x: 0 ✓, y: 0 ✓ , z: 1 ✓</div> <div>#include <stdio.h> int main(void) { int x=0, y=0, z=1; if (z<x y>=z && z==1) if (z&& y) y=1; else x=1; printf("x:%d, y:%d, z:%d\n", x, y, z); return 0; }</div> | <div>x: 1 ✓, y: 0 ✓ , z: 0 ✓</div> <div>#include <stdio.h> int main(void) { int x=0, y=0, z=1; if (z=y) { y++; z--; } else ++x; printf("x:%d, y:%d, z:%d\n", x, y, z); return 0; }</div> | <div>x: 0 ✓, y: 1 ✓ , z: 0 ✓</div> <div>#include <stdio.h> int main(void) { int x=0, y=0, z=1; if (z=x<y) { x += 3; y -= 1; } else x=y++; printf("x:%d, y:%d, z:%d\n", x, y, z); return 0; }</div> |
|---|---|--|

문제 17

부분적으로 맞음

총 5.00 점에서
4.00 점 할당

What is the output of the following program?

#include <stdio.h>
int main(void)
{
 int i; // index
 int len; // length
 char msg[]="Hello 0.\0.7\n";

 for(i=0;msg[i]!='\0';i++)
 ;
 len = i;

 printf("The size of the char array msg[] is %d\n",sizeof(msg));
 printf("The length of the string [%s] : %d\n", msg,len);

 return 0;
}

Execution Result

The size of the char array msg[] is 11 ✖

The length of the string [Hello 0.] : 8 ✓

문제 18

정답

총 5.00 점에서
5.00 점 할당

What is the output of the following program ?

- Note that we can use a hexadecimal constant in the C language by adding "0x" in front of an hexadecimal value as shown in the variable initialization part the of the code.
- The format specifier "%X" in printf() converts an unsigned integer into hexadecimal representation.

Execution Result

```

un1=0x1234
un2=0xABCD
un1&mask=0x 1200 ✓
un1|mask=0x FF34 ✓
un1^un1=0x 0000 ✓
un1^0=0x 1234 ✓
(un1^un2)^un2=0x 1234 ✓

```

```

#include <stdio.h>
int main(void)
{
    unsigned short un1 = 0x1234;        // Hexadecimal Value
    unsigned short un2 = 0xABCD;        // Hexadecimal Value
    unsigned short mask = 0xFF00;       // Hexadecimal Value

    printf("un1=0x%X\n", un1);
    printf("un2=0x%X\n", un2);

    printf("un1&mask=0x%X\n", un1&mask);
    printf("un1|mask=0x%X\n", un1|mask);

    printf("un1^un1=0x%X\n", un1^un1);
    printf("un1^0=0x%X\n", un1^0);

    printf("(un1^un2)^un2=0x%X\n", (un1^un2)^un2);

    return 0;
}

```

문제 19

정답

총 5.00 점에서
5.00 점 할당

The user input value of variable a is 3. What is the output of the following source code

? 2 ✓

| | output |
|---|-------------------|
| 1 | a==3, |
| 2 | a==3, a==4, |
| 3 | a==2, a==3, |
| 4 | a==2, a==3, a==4, |

```
#include <stdio.h>
int main(void) {
    int a;

    scanf("%d",&a);
    switch( a ) {
        case 1:
            printf("a==1, ");
            break;
        case 2:
            printf("a==2, ");
        case 3:
            printf("a==3, ");
        case 4:
            printf("a==4, ");
            break;
        case 5:
            printf("a==5, ");
        default:
            printf("a==other, ");
    }
    printf("\n");

    return 0;
}
```

문제 20

정답

총 4.00 점에서
4.00 점 할당

For each "case" statements in the following source code, determine whether it raises a syntax error or not. if an error, select 'E'; otherwise select 'X'

- case 1: X ✓
- case two: E ✓
- case 1+2: X ✓
- case FOUR: X ✓

```
#include <stdio.h>
#define FOUR 4
int main(void) {
    int a, two=2;

    scanf("%d", &a);
    switch( a ) {
        case 1: // 1
            printf("a==one\n");
            break;
        case two: // 2
            printf("a==two\n");
            break;
        case 1+2: // 3
            printf("a==three\n");
            break;
        case FOUR: // 4
            printf("a==four\n");
            break;
        default:
            printf("a==other\n");
    }
    return 0;
}
```

문제 21

정답

총 5.00 점에서
5.00 점 할당

What is the execution result of the following source code?

2 ▼



1. i=2 i=3
2. i=3 i=3 i=4
3. i=3 i=4
4. i=2 i=3 i=4

```
#include <stdio.h>
int main(void) {
    int i=2;

    while (i < 3) {
        printf("i=%d ", ++i);
    }

    do {
        printf("i=%d ", i++);
    } while (i < 3);

    printf("i=%d\n", i);
    return 0;
}
```