프로그래밍1및실습 기말고사 모범답안

I. 객관식 문제 (16문항, 문항당 5점 / 총 80점)

1. 다음 함수 사용법 중 권장되지 않는 것은?

(1) int add(int a, int b)

{

int result = a + b;

return result;

}

(2) void print(int a, int b)

{

printf("%d\n", a+b);

}

(3) void print(int a, int b)

{

printf("%d\n", a+b);

return;

}

(4) void find\_seven()

{

int i;

for(i=0; i<10; i++) {

if(i==7) return;

printf("%d\n", i);

}

}

**(5) int div(int a, int b)**

**{**

**double result = (double)a / b;**

**return result;**

**}**

2. 다음 함수가 가진 특성을 잘 설명한 것은?

|  |
| --- |
| long long factorial(int x)  {  if(x <= 1) return 1;  return x \* factorial(x-1);  } |

(1) 함수는 자기자신을 반복해서 호출할 수 없으므로 위 프로그램은 잘못되었다.

**(2) 탈출조건이 없으면 무한 호출되므로 프로그램이 제대로 작동하지 않는다.**

(3) 위의 factorial() 과 return문에 있는 factorial() 은 다른 함수이다.

(4) x의 값에 관계없이 자기자신을 호출한다.

(5) factorial(x) 와 factorial(x-1) 의 값은 x에 관계없이 항상 같다.

3. 다음 프로그램에서 컴파일되는 문장이 아닌 것은?

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #define USE  int main()  {  int val;  #ifndef USE  sscanf("123", "%d", &val);  #else  val = atoi("123");  #endif  printf("Val = %d\n", val);  return 0;  } |

**(1) sscanf("123", "%d", val);**

(2) int val;

(3) val = atoi("123");

(4) printf("Val = %d\n", val);

(5) return 0;

4. 다음 프로그램에서 ①의 위치에서 b의 값은?

|  |
| --- |
| #include <stdio.h>  int b = 10;  void blahblah() {  static int b = 123;  printf("b = %d\n", b);  b++;  }  int main()  {  int i;  for(i=0; i<10; i++) {  blahblah();  b++;  }  printf("b = %d\n", b);  blahblah(); -----①  return 0;  } |

**(1) 20**

(2) 123

(3) 133

(4) 0

(5) 143

5. short \*ps 가 0x60FE10 번지를 가리키고 있을 때 ps-7은 어디를 가리키는가?

(1) 0x60FE17

(2) 0x60FE09

(3) 0x60FE1E

**(4) 0x60FE02**

(5) 0x60FE10

6. 다음 프로그램에서 빈 칸에 알맞은 문자열은?

|  |
| --- |
| #include<stdio.h>  int main()  {  int arr[100];  FILE \*fp = fopen("test.dat", ( )); // 파일을 읽어온다.  for(int i=0; i<100; i++) {  fscanf("%d\n", &arr[i]);  }  fclose(fp);  } |

(1) "w"

(2) "wb"

**(3) "r"**

(4) "rb"

(5) "r+b"

7. 다음 중 구조체 포인터에 대한 설명으로 틀린 것은?

(1) 포인터 ps가 가리키는 구조체의 멤버 abc를 참조하려면 ps->abc 로 기술한다.

(2) 가리키는 구조체의 첫번째 멤버의 주소이다.

(3) 구조체의 크기가 크다면 함수의 매개변수로 구조체 포인터를 쓰는 것이 좋다.

(4) 자기자신의 타입으로 된 구조체를 멤버로 삼기 위해 사용할 수 있다.

**(5) 포인터 ps가 가리키는 구조체의 멤버 abc를 참조하려면 \*ps.abc 로 기술한다.**

8. 다음 중 입출력장치와 프로그램 간의 데이터 연결 경로를 의미하는 용어는?

(1) 데이터

**(2) 스트림**

(3) 버퍼

(4) 배열

(5) 메모리

9. 다음 중 printf문에서 포인터의 값(메모리 주소)을 출력하기 위한 형식지정자는?

(1) %d

(2) %f

(3) %s

**(4) %p**

(5) %c

10. 다음 중 아래 문장들을 모두 실행한 상태에서 ptr+3 을 표현한 것은?

|  |
| --- |
| int arr[10] = { 0, };  int \*ptr;  ptr = arr;  ptr = ptr+2; |

**(1) &arr[5]**

(2) arr[5]

(3) arr

(4) arr[0]

(5) &arr[0]

11. 다음 중 int main(int argc, char \*argv[]) 에 대한 설명으로 틀린 것은?

(1) main 함수의 반환형은 int이다.

(2) argc는 명령행 인자의 수이다.

(3) argv는 명령행 인자(문자열) 의 배열이다.

**(4) argv[0] 은 실행파일명 뒤의 첫 번째 명령행 인자이다.**

(5) 실행파일명을 포함해 명령행 인자가 3개이면 argc는 3이다.

12. 다음 중 반복문에 대한 설명으로 틀린 것은?

(1) for문의 초기식, 조건식, 증감식은 생략 가능하다.

(2) for문은 while문으로 고쳐 적을 수 있다.

(3) for문에서 초기식은 처음 한 번만 실행된다.

(4) while문 안의 조건식에 0 또는 1을 적을 수 있다.

**(5) do … while 문에서 while문의 조건식이 처음부터 만족하지 않으면 안의 내용이 아예 실행되지 않는다.**

13. 다음 중 문자열(문자 배열)에 대한 설명으로 올바른 것은?

(1) C언어에서 문자열은 별개의 데이터형이다.

**(2) 문자열이 끝났음을 표시하기 위하여 문자열 내용 뒤에 '\0' 이 추가되어야 한다.**

(3) str이 문자열 "Hello" 이면 strlen(str) 의 값은 뒤의 '\0' 을 포함하여 6이다.

(4) 문자열을 초기화하는 방법은 일반적인 배열 초기화 방법밖에 없다.

(5) str이 문자열 "World!" 이면 sizeof(str) 은 문자의 개수인 6이다.

14. 아래 구조체 선언 및 초기화 후 std.c의 값은?

|  |
| --- |
| struct Student {  char a[20];  int b;  char c[20];  int d;  };  ...(생략)  struct Student std = { "철수", 20029614, "컴퓨터공학과", 88 }; |

(1) "철수"

(2) 88

**(3) 컴퓨터공학과**

(4) 20029614

(5) { "철수", 20029614, "컴퓨터공학과", 88 }

15. 다음 중 파일 입출력 함수에 대한 설명으로 틀린 것은?

(1) fprintf(stdout, "Hello, world!"); 는 printf("Hello, world!"); 와 같다.

**(2) fread 함수와 fwrite 함수는 텍스트 파일을 다루는 데 사용할 수 없다.**

(3) fscanf 함수는 파일의 내용을 입력 형식에 맞춰 읽어들인다.

(4) 키보드에서 문자열을 입력해서 받을 때 gets 함수보다는 fgets 함수를 사용하는 것이 좋다.

(5) 파일의 특정 위치부터 읽거나 쓰기 위해 fseek 함수를 사용한다.

16. 다음 중 배열 array의 10번째 원소의 위치를 가리키는 표현은?

(1) array

(2) array[10]

(3) &array[10]

**(4) &array[9]**

(5) array[9]

II. 단답형 주관식 문제 (10문항, 문항당 8점 / 총 80점)

1. C의 표준 입출력 스트림 두 가지를 적으시오.

**stdin, stdout**

2. 다음 프로그램과 실행결과를 보고 빈 칸에 알맞은 내용을 적으시오.

|  |
| --- |
| #include <stdio.h>  int main()  {  for(int i=0; i<10; i++) {  if(i == 5) **( )**  printf("Hello, %d\n", i);  }  } |
| Hello, 0  Hello, 1  Hello, 2  Hello, 3  Hello, 4  Hello, 6  Hello, 7  Hello, 8  Hello, 9 |

**continue;**

3. 파일의 종류 중 데이터의 형태에 따른 종류 두 가지를 적으시오.

**텍스트 데이터, 바이너리 데이터**

4. 아래 문장에서 밑줄 친 부분을 무엇이라고 하는가?

|  |
| --- |
| int sub(int a, int b)  {  return a - b;  }  int main()  {  printf("%d\n", sub(**4,2**));  } |

**(함수의) 실매개변수**

5. 다음 밑줄 친 부분을 가리키는 변수의 존재범위에 따른 종류는?

|  |
| --- |
| #include <stdio.h>  static int value;  int main()  {  for(int i=0; i<100; i++) {  printf("value = %d\n", value);  }  } |

**외부 정적 변수**

6. 배열을 포인터의 관점에서 어떻게 표현할 수 있는가?

**포인터 상수**

7. 다음과 같이 설명되는 용어를 적으시오.

|  |
| --- |
| 변수를 함수에 전달할 때 값이 아닌 변수를 가리키는 데이터를 전달 |

**참조에 의한 호출**

8. 다음 설명대로 소스 프로그램을 변환하는 프로그램을 무엇이라고 하는가?

|  |
| --- |
| 컴파일하기 전에 먼저 처리해야 할 작업을 수행 |

**전처리기**

9. 다음 설명이 의미하는 것은 무엇인가?

|  |
| --- |
| * 개발자 혹은 제3자가 만든 기능을 사용하기 위해 포함하는 파일 * 프로그램 구현 내용보다는 어떤 함수나 상수를 사용할 수 있는 지만 표기하는 파일 |

**헤더파일**

10. scanf("%d", &var); 에서 &var의 의미는 무엇인가?

**변수 var의 (메모리) 주소 (또는 위치)**

III. 서술형 주관식 (6문항, 문항당 15점 / 총 90점)

1. 다음 프로그램이 상식에 어긋나게 실행되는 이유를 설명하고 잘못된 부분을 올바르게 고치시오.

|  |
| --- |
| #include <stdio.h>  #define CUBE(x) x\*x\*x  int main()  {  printf("=== Cube table ===\n");  for(int i=0; i<10; i++) {  printf("(%d+1)^3 = %d\n", i+1, CUBE(i+1));  }  } |

**매크로 함수는 표현 그대로를 치환하기 때문에 CUBE(i+1) 은 i+1\*i+1\*i+1 로 치환되며, 연산자 우선 순위에 따라서 i+(1\*i)+(1\*i)+1 로 계산하여 의도한 대로 세제곱수를 제대로 계산하지 못한다. 따라서 이를 올바르게 표현하려면 #define CUBE(x) x\*x\*x 을 #define CUBE(x) ((x)\*(x)\*(x)) 로 고쳐야 한다. (뒤의 부분을 (x)\*(x)\*(x) 로 해도 정답)**

**또 다른 정답: printf문에 있는 CUBE(i+1) 을 CUBE((i+1)) 로 수정**

**고치는 방식을 아래와 같이 함수로 해도 정답**

**int CUBE(x)**

**{**

**return x\*x\*x;**

**}**

2. 다음은 문자열이 같은지 비교하고 같으면 Same string을 출력하고 끝나는 프로그램이다. 잘못된 부분을 찾고 그 이유를 간단히 적고 어떻게 고쳐야 할 지 서술하시오.

|  |
| --- |
| #include<stdio.h>  int main()  {  char \*str[] = "hello";  char \*cp = "hello";    while(cp = str) {  printf("Same string.\n");  }  return 0;  } |

**잘못된 곳 1: char \*str[]**

**수정 1: char str[] (문자열은 문자 배열이지 문자 배열의 포인터가 아니다.)**

**잘못된 곳 2: cp = str**

**수정 2: strcmp(cp, str) == 0 (문자열 비교는 = 가 아니라 strcmp 함수를 써야 한다.)**

**잘못된 곳 3: while**

**수정 3: if (while로 하면 무한루프이기 때문에 프로그램이 끝나지 않는다.)**

3. 배열과 구조체의 차이점을 서술하시오.

**배열: 같은 자료형 데이터를 여러 개 담는 구조**

**구조체: 서로 다른 자료형의 값들을 하나의 범주로 표현하기 위한 데이터 구조**

4. 1부터 n까지 int형 숫자를 누적해서 곱하는 누적곱을 구하는 함수를 구현한다. 반환형은 int이며, 재귀함수 형식은 사용할 수 없다. 그리고 구현한 함수를 이용하여 main() 에서 1부터 10까지 누적곱의 값을 출력하는 프로그램을 서술하시오. 단, 누적곱 구하는 함수에서 매개변수에 0 이하의 숫자가 오는 경우 1을 반환하도록 하고, 누적곱의 값을 출력할 때에는 각 숫자가 붙지 않고 구별되게 표시되도록 하시오.

**(정답 예시. 올바르게 구현한 프로그램은 모두 정답)**

**int factorial(int x)**

**{**

**int sum=1;**

**if(x <= 0) return 1;**

**for(int i=1; i<=x; i++) {**

**sum \*= i;**

**}**

**return sum;**

**}**

**int main()**

**{**

**for(int i=1; i<=10; i++) {**

**printf(“%d\n”, factorial(i));**

**}**

**}**

(문제에서 요구하는 답을 적으면 모두 OK.)

5. 1부터 15까지 sum() 의 반환값을 sum.txt 에 기록하는 프로그램을 작성할 때 빈 칸의 내용을 채우시오. (단, 파일 오류 처리는 고려하지 않음)

|  |  |
| --- | --- |
| **#include<stdio.h>**  **int sum(int x)**  **{**  **int result=0;**  **if(x <= 0) return 0;**  **for(int i=1; i<=x; i++) {**  **result += I;**  **}**  **return result;**  **}**  **int main()**  **{**   |  | | --- | |  |   **}** |

**(정답 예시. 같은 결과가 나오면 어떻게 적든 OK. 파일 오류 처리는 있어도 없어도 상관없음.)**

**FILE \*fp;**

**int i;**

**fp = fopen("sum.txt", "w");**

**for(i=1; i<=15; i++) {**

**fprintf(fp, "%d\n", sum(i));**

**}**

**fclose(fp);**

6. 지역 변수와 내부 정적 변수의 차이점을 서술하시오.

**지역변수**

* **존재 위치**
  + **main() 안에**
  + **사용자 정의 함수 안에**
  + **조건문이나 반복문 안에(초기식에 선언된 경우 포함)**
  + **기타 사용자가 지정한 블록 { } 안에**
* **지역 변수는 함수나 조건문/반복문, 기타 블록을 벗어나면 더 이상 존재하지 않고 사라짐**
* **존재 범위를 벗어나는 곳에서 접근하려고 하면 정의되지 않았다고 오류 발생**

**내부 정적 변수**

* **블록 내에서 static 키워드를 이용하여 선언하는 정적 변수**
* **블록을 벗어나면 존재하지 않다가 블록 내부로 들어가면 존재가 드러남**
* **일반 지역 변수와 달리 블록을 벗어나도 값을 보존함**
* **블록이 지속적으로 존재해야 하므로 현실적으로 함수 내에서 사용되는 것이 일반적**