|  |
| --- |
| 1일차 C언어 심화 워크샵 솔루션 |

* **답안 제출은 A4용지에 수기로 작성하여 제출하시오.**

**[ 1과. 수의 표현 ]**

1. **다음 코드의 인쇄 결과는?(10p)**

|  |
| --- |
| **signed int a = 0xffffffff;**  **unsigned int b = -1;**  **printf("%d, %u, %x\n", a, a, a);**  **printf("%d, %u, %x\n", b, b, b);** |

**🡺 -1, 4294967295, ffffffff**

**🡺 -1, 4294967295, ffffffff**

1. **다음 코드의 인쇄 결과는?(11p)**

|  |
| --- |
| **signed char sc = 0xff;**  **unsigned char uc = -1;**  **char c = uc;**  **printf("%x, %x, %x", sc, uc, c);** |

**🡺 ffffffff, ff, ffffffff**

1. **다음 코드의 인쇄 결과는?(14p)**

|  |
| --- |
| **signed int a = -1;**  **unsigned int b = a;**  **printf("%d, %u, %x\n", a, a, a);**  **printf("%d, %u, %x\n", b, b, b);** |

**🡺 -1, 4294967295, ffffffff**

**🡺 -1, 4294967295, ffffffff**

1. **다음 코드의 인쇄 결과는?(15p)**

|  |
| --- |
| **signed char sc = 127;**  **unsigned char uc = 1;**  **char c = sc + uc;**  **printf("%d", c);** |

**🡺 -128**

1. **다음 코드의 인쇄 결과는?(15p)**

|  |
| --- |
| **short ss = 65535 - 1;**  **printf("%hd, %hu, %hx\n", ss, ss, ss );** |

**🡺 -2, 65534, fffe**

1. **다음 코드의 인쇄 결과는?(26p)**

|  |
| --- |
| **float a=3.25;**  **double b=3.25;**  **if( a == b ) printf("true");**  **else printf("false");** |

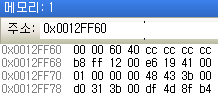
**🡺 true**

1. **변수 f, d의 메모리에 저장된 16진수 값을 각각 구하시오.(31p)**

|  |
| --- |
| **float f=3.625f;**  **double d=3.625;** |

**🡺 0x40680000, 0x400d000000000000**

**❑ 다음 메모리 DUMP 결과를 보고 다음을 구하시오.(8~12번)**

****

1. **int a; 에서 &a는 0x0012FF6C일 때 printf(“%x\n”, a); 결과는?**

**🡺 4119e6**

1. **unsigned short s; 에서 &s는 0x0012FF6A일 때 printf(“%x”, s); 결과는?**

**🡺 12**

1. **signed char c; 에서 &c는 0x0012FF70일 때 printf(“%x”, c); 결과는?**

**🡺 1**

1. **float f; 에서 &f는 0x0012FF60일 때 16진수로 값은?**

**🡺 0x40600000**

1. **double d; 에서 &d는 0x12FF74일 때 16진수로 값은?**

**🡺 0x003b31d0003b4348**

**[ 2과. 데이터의 형변환 ]**

1. **형 변환에 관한 질문에 빈칸을 채우시오.(41p)**
2. **int보다 작은 정수타입은 무조건 ( )로 승격된다.**
3. **singed int와 unsigned int는 ( )를 크게 본다.**
4. **type이 다른 두 요소가 연산을 하는 경우 작은 타입은 ( )타입에 맞춘다.**
5. **우변이 타입이 다른 좌변으로 대입되는 경우 ( )의 타입이 ( )의 타입으로 변경된다.**

**🡺 singed int / unsigned int / 큰 / 우변 / 좌변**

1. **다음 코드의 실행결과를 예측하고 메모리로 그려보시오.(42p)**

|  |
| --- |
| **signed int si=0x12345678; unsigned int ui=0x12345678;**  **signed short ss; unsigned short us;**  **ss=si; us=si;**  **printf("%x, %x\n", ss, us);**  **ss=ui; us=ui;**  **printf("%x, %x\n", ss, us);** |

**🡺 5678, 5678 / 5678, 5678**

1. **다음 코드의 실행결과를 예측하고 연산과정을 메모리로 그려보시오.(47p)**

|  |
| --- |
| **unsigned char uc=0xff;**  **signed char sc=0xff;**  **int a, b;**  **a = uc + 1; b = sc + 1;**  **printf("%d, %d\n", a, b);** |

**🡺 256, 0**

1. **다음 코드의 실행결과를 예측하고 메모리를 그려보시오 (47p)**

|  |
| --- |
| **signed char sc = -1;**  **unsigned char uc = -1;**  **printf(" %x %x ", sc + sizeof( char ), uc + sizeof( char ) );** |

**🡺 0, 100**

1. **Cast연산자를 이용하여 밑줄 친 문장을 단 하나의 경고도 없도록 수정하시오.(48p)**

|  |
| --- |
| **double d=3.5;**  **float f;**  **f = d + 0.5;**  **printf(“%f\n”, f );** |

**🡺 f= (float)(d + 0.5);**

1. **다음 코드의 인쇄 결과는?(50p)**

|  |
| --- |
| **signed int a= -1;**  **unsigned char b= -1;**  **if( a<b ) printf("unsigned \n");**  **else if( a>b ) printf("singed\n");**  **else printf("same");** |

**🡺 unsigned**

**[ 3.1과 무한확장형 변수의 해석 ]**

1. **다음 보기의 변수들의 분석그래프, 작명등을 구하시오. ( (1) ~ (4)번까지 )**

|  |
| --- |
| **[1] int a;**  **[2] int a[4];**  **[3] int \* p;**  **[4] int a[2][3];**  **[5] int \*\*p;**  **[6] int \*a[3];**  **[7] int \*\*a[3];**  **[8] int (\*p)[3];**  **[9] int \*(\*p)[3];**  **[10] int (\*\*p)[3];**  **[11] int (\*a[2])[3];**  **[12] int \*(\*a[2])[3];**  **[13] int \* f(void);**  **[14] int \*(\*p)(void);**  **[15] int \*(\*a[3])(void);**  **[16] struct st \*(\*a[3])(int \*);**  **[17] int (\*f(void))[3];**  **[18] int \*(\*f(void))[3];**  **[19] void (\*f(void))(void);**  **[20] int \*(\*f(int\*))(int\*);** |

1. **<보기>선언들의 분석그래프를 그리시오.(64p)**



1. **<보기>선언들의 이름을 만드시오.(65p)**

**예) int \*p 🡺 int 포인터 p**

**[1] int a**

**[2] int 4배열 a**

**[3] int 포인터 p**

**[4] int 3배열 2배열 a**

**[5] int 포인터 포인터 p, int 이중포인터 p**

**[6] int 포인터 3배열 a**

**[7] int 포인터 포인터 3배열 a, int 이중포인터 3배열 a**

**[8] int 3배열 포인터 p**

**[9] int 포인터 3배열 포인터 p**

**[10] int 3배열 포인터 포인터 p, int 3배열 이중포인터 p**

**[11] int 3배열 포인터 2배열 a**

**[12] int 포인터 3배열 포인터 2배열 a**

**[13] int 포인터를 리턴하는 함수 f**

**[14] int 포인터를 리턴하는 함수 포인터 p**

**[15] int 포인터를 리턴하는 함수 포인터 3배열 a**

**[16] struct st 포인터를 리턴하는 함수 포인터 3배열 a**

**[17] int 3배열 포인터를 리턴하는 함수 f**

**[18] int 포인터 3배열 포인터를 리턴하는 함수 f**

**[19] 리턴이 없는 함수 포인터를 리턴하는 함수 f**

**[20] int 포인터를 리턴하는 함수 포인터를 리턴하는 함수 f**

1. **<보기>선언들의 타입과 identifier에 &와 \*을 붙인 타입을 구하시오. (72p)**

**(단. 함수의 identifier에 \*을 붙이는 것은 제외🡺 추후 배움)**

**예) int \*p : p 🡺 int\*, &p 🡺 int \*\*, \*p 🡺 int**

|  |  |  |
| --- | --- | --- |
| **[1] a 🡺 int**  **[2] a 🡺 int [4], int \***  **[3] p 🡺 int \***  **[4] a 🡺 int [2][3], int(\*)[3]**  **[5] p 🡺 int \*\***  **[6] a 🡺 int \*[3]**  **[7] a 🡺 int \*\*[3] , int \*\*\***  **[8] p 🡺 int (\*)[3]**  **[9] p 🡺 int \*(\*)[3]**  **[10] p 🡺 int (\*\*)[3]**  **[11] a🡺int(\*[2])[3], int(\*\*)[3]**  **[12] a 🡺 int \*(\*[2])[3], int \*(\*\*)[3]**  **[13] f 🡺 int \* (void)**  **[14] p 🡺 int \*(\*)(void)**  **[15]a🡺**  **int \*(\*[3])(void), int\*(\*\*)(void)**  **[16] a🡺 struct st \*(\*[3])(int \*), struct st \*(\*\*)(int \*)**  **[17] f 🡺 int (\*(void))[3]**  **[18] f 🡺 int \*(\*(void))[3]**  **[19] f 🡺 void (\*(void))(void)**  **[20] f 🡺 int \*(\*(int\*))(int\*)** | **[1] &a 🡺 int \***  **[2] &a 🡺 int (\*)[4]**  **[3] &p 🡺 int \*\***  **[4] &a 🡺 int (\*)[2][3]**  **[5] &p 🡺 int \*\*\***  **[6] &a 🡺 int \*(\*)[3]**  **[7] &a 🡺 int \*\*(\*)[3]**  **[8] &p 🡺 int (\*\*)[3]**  **[9] &p 🡺 int \*(\*\*)[3]**  **[10] &p 🡺 int (\*\*\*)[3]**  **[11] &a 🡺 int (\*(\*)[2])[3]**  **[12] &a 🡺 int \*(\*(\*)[2])[3]**  **[13] &f 🡺 int \*(\*) (void)**  **[14] &p 🡺 int \*(\*\*)(void)**  **[15] &a🡺 int \*(\*(\*)[3])(void)**  **[16] &a 🡺**  **struct st \*(\*(\*)[3])(int \*)**  **[17] &f 🡺 int (\*(\*)(void))[3]**  **[18] &f 🡺 int \*(\*(\*)(void))[3]**  **[19]&f🡺void (\*(\*)(void))(void)**  **[20] f 🡺 int \*(\*(\*)(int\*))(int\*)** | **[1] \*a 🡺 ERROR**  **[2] \*a 🡺 int**  **[3] \*p 🡺 int**  **[4] \*a 🡺 int [3] , int \***  **[5] \*p 🡺 int \***  **[6] \*a 🡺 int \***  **[7] \*a 🡺 int \*\***  **[8] \*p 🡺 int [3], int \***  **[9] \*p 🡺 int \*[3], int \*\***  **[10] \*p 🡺 int (\*)[3]**  **[11] \*a 🡺 int (\*)[3]**  **[12] \*a 🡺 int \*(\*)[3]**  **[13] \*f 🡺 추후**  **[14] \*p 🡺 int \* (void)**  **[15] \*a🡺 int \*(\*)(void)**  **[16]\*a🡺**  **struct st \*(\*)(int \*)**  **[17] \*f 🡺 추후**  **[18] \*f 🡺 추후**  **[19] \*f 🡺 추후**  **[20] \*f 🡺 추후** |

1. **<보기>선언들의 타입을 가르키는 포인터변수 q, 요소로 하는 배열 b[2], 리턴 하는 함수 g(void)를 선언하시오. (76p)**

**예) int \*p : int \*\*q, int \*b[2], int \* g(void)**

|  |  |  |
| --- | --- | --- |
| **[1] int \*q;**  **[2] int (\*q)[4];**  **[3] int \*\*q;**  **[4] int (\*q)[2][3];**  **[5] int \*\*\*q;**  **[6] int \*(\*q)[3];**  **[7] int \*\*(\*q)[3];**  **[8] int (\*\*q)[3];**  **[9] int \*(\*\*q)[3];**  **[10] int (\*\*\*q)[3];**  **[11] int (\*(\*q)[2])[3];**  **[12] int \*(\*(\*q)[2])[3];**  **[13] int \* (\*q)(void);**  **[14] int \*(\*\*q)(void);**  **[15] int \*(\*(\*q)[3])(void);**  **[16]**  **struct st \*(\*(\*q)[3])(int \*);**  **[17] int (\*(\*q)(void)[3];**  **[18] int \*(\*(\*q)(void)[3];**  **[19] void (\*(\*q)(void))(void);**  **[20] int \*(\*(\*q)(int\*))(int\*);** | **[1] int b[2];**  **[2] int b[2][4];**  **[3] int \* b[2];**  **[4] int b[2][2][3];**  **[5] int \*\*b[2];**  **[6] int \* b[2][3];**  **[7] int \*\* b[2][3];**  **[8] int (\*b[2])[3];**  **[9] int \*(\*b[2])[3];**  **[10] int (\*\*b[2])[3];**  **[11] int (\*b[2][2])[3];**  **[12] int \*(\* b[2][2])[3];**  **[13] ERROR**  **[14] int \*(\*b[2])(void);**  **[15] int \*(\*b[2][3])(void);**  **[16]**  **struct st \*(\*b[2][3])(int \*);**  **[17] ERROR**  **[18] ERROR**  **[19] ERROR**  **[20] ERROR** | **[1] int g(void);**  **[2] ERROR**  **[3] int \* g(void);**  **[4] ERROR**  **[5] int \*\* g(void);**  **[6] ERROR**  **[7] ERROR**  **[8] int (\*g(void))[3];**  **[9] int \*(\*g(void))[3];**  **[10] int (\*\*g(void))[3];**  **[11] ERROR**  **[12] ERROR**  **[13] ERROR**  **[14] int \*(\*g(void))(void);**  **[15] ERROR**  **[16] ERROR**  **[17] ERROR**  **[18] ERROR**  **[19] ERROR**  **[20] ERROR** |