

1. 다음 C 프로그램의 출력을 써라.

(단, 답만 쓰면 점수가 없고, 메모리상의 32-bit 내용을 binary로 표시하고, 출력 결과를 써라.)

```
#include <stdio.h>
union Data{
    float k ;
    unsigned int a;
} data;

int main() {

    data.k = 4.25 ;
    printf("%x \n", data.a) ; /* (a) 5점 */
                                4088000

    data.k = -4.25 ;
    printf("%x \n", data.a) ; /* (b) 5점 */
                                c088000

    return(0) ;
}
(10점)
```

2. 다음 C 프로그램을 cspro에서 gcc를 이용하여 수행시켰을 때 출력 결과를 써라.

단, (e) ~ (h)는 계산 식도 함께 써야 점수를 받을 수 있음. (각 3점)

```
#include <stdio.h>

char aaa ;
enum days {Mon, Tue, Wed, Thu};

char bbb[5][10][20]={ 'a', 'bb', '\0' };

struct aa {
    float j; char aaa; char bbb; enum days bb;
} aaaa;

union utag {
    char bb; float kk ; struct aa lala; char *c ;
} haha, test[10][6] ;

void main() {
    printf("%d \n", sizeof(char *)); /* Result : 8 */
    printf("%d \n", sizeof(int)); /* Result : 4 */
    printf("%d \n", sizeof(float)); /* Result : 4 */
    printf("%d \n", &(test[0][0])); /* Result : 6768 */
    printf("%d \n", sizeof(haha)); /* Result : 16 */

    printf("%d \n", sizeof(bbb)); /* (a) */ 1000
    printf("%d \n", strlen(bbb)); /* (b) */
    printf("%d \n", sizeof(aaaa)); /* (c) */ 12
    printf("%d \n", sizeof(test)); /* (d) */ 720

    printf("%d \n", &(test[3][4].lala.bb)); /* (e) */ 7040
    printf("%d \n", &(test[3][4].lala)+1); /* (f) */ 7044
    printf("%d \n", &(test[4][11])); /* (g) */ 7188
    printf("%d \n", &(test[4][11])+1); /* (h) */ 7200
}
주소의 타입을 보고 + 스케일링
```

(24점)

3. 프로그래밍 언어에서의 binding은 여러 시점에서 일어난다.

다음과 같은 binding이 일어나는 시점은 각각 언제인가 ? (각 3점 x 8 = 24점)

- (a) local 변수의 storage 바인딩 **runtime**
  - (b) C 언어에서 포인터 변수의 타입 바인딩 **컴파일타임**
  - (c) 함수의 호출과 그에 대응되는 수행 코드의 바인딩 **링킹타임**
  - (d) structure 변수의 storage 크기 바인딩 **runtime**
  - (e) named constant의 value 바인딩 **로딩타임**      서브루틴 정의 변수들
  - (f) C 언어에서 formal parameter에 대한 value 바인딩 **런타임 - 프로시저 호출될때**
  - (g) malloc()에 의하여 할당된 변수의 타입 바인딩 **타입은 컴파일**
  - (h) Set 변수에 대한 storage size 바인딩 **스토리지는 런타임**
- (24점)      **비트수**  
**element 개수가 몇개인지 알아야함 컴파일타임**

4. 아래와 같은 C-like 언어에서 (a) static scoping rule을 사용하는 경우와 (b) dynamic scoping rule을 사용하는 경우에 각각 print되는 값은 무엇인가 ? (각 5점 x 2 = 10점)

```
int a, b ;

int p(void) {
    int a, p ;
    a = 0 ; b = 1 ; p = 2;
    return(p) ;
}

void print(void) {
    printf("a=%d,b=%d\n",a,b);
}

void q(void) {
    int b ;
    a = 3 ; b = 4 ;
    print();
}

void main(void) {
    a = p() ;    3    1
    q() ;        3    4
}
```

(10점)

5. 어떤 같은 문제를 해결하기 위하여 C와 FORTRAN과 같은 Imperative Language로 작성한 프로그램과 PROLOG나 LISP과 같은 declarative 언어로 작성한 2개의 프로그램이 있다고 가정하자. 이 두 프로그램을 LINUX가 설치된 Pentium PC에서 똑 같이 컴파일한 후에 수행시켰을 때 일반적으로 어떤 언어로 작성한 프로그램의 수행 속도가 빠른가 ? 그 이유를 "semantic gap" 관점에서 설명하라. (10점)  
(10점)

c, fortran은 컴파일이 빠름. 하드웨어 바탕언어 gap이 작음  
prolog lisp는 하드웨어랑 상관 없어서 컴파일 시간이 걸림 gap이 큼  
semantic gap - language가 돌아가는 semantic과 하드웨어 매커니즘의 차이  
기계언은 gap이 거의 없음

6. 신뢰도 있는 프로그램의 작성을 도와주기 위하여 프로그래밍 언어가 제공하여야 하는 기능들은 어떤 것들이 있는가 ? 대표적인 기능 3개 및 이런 기능을 제공하기 위한 방법에 대하여 설명하라. (각 2점 x 3 = 6점)  
(6점)

1. type checking  
2. exception handling divide by z, heap없음 등이 일어났을때 어떤 일을할지 정함  
3. range checking 런타임 때 배열을 벗어나는가  
4. alias 금지 포인터 금지  
5. goto 금지

7. 아래의 C 프로그램을 이용하여 다음의 물음에 답하라. (11점)

```
#include <stdio.h>
int two[2][] = {1,2,3,4};
main() {
    char ch = 'C'; /* 'C'의 ASCII code는 67이다. */
    int i; float f;
    f = i = ch + 1;
    printf("%c ", i); /* (b) */
    printf("%d, %f\n", i, f); /* (c) */
}
```

- (a) 위의 프로그램을 컴파일하면 error가 난다. Compile error가 나는 곳을 지적하고, 그 이유를 설명하라. (5점)  
(b)-(c) error가 없도록 수정한 후 이 프로그램에 의하여 프린트 되는 내용을 써라.  
(각 3점 x 2 = 6점)  
(11점)

two.  
크기를 지정안해줌

D 68 68.000