

1. 프로그래밍 언어 일반 (8점)

- (a) 과학계산 응용에서 많이 사용되는 matrix는 프로그래밍 언어에서 2D array로 표현할 수 있다. 이런 기능은 FORTRAN이나 C 언어에서 모두 표현할 수 있지만, 이 2개의 언어 중에서 FORTRAN을 과학계산용 언어라 할 수 있는 이유를 구체적인 FORTRAN의 기능 예를 들어 설명하라. (4점)
- (b) Java 언어로 작성된 프로그램은 여러 종류의 platform상에서 Web 브라우저만 있으면 재컴파일할 필요 없이 수행시킬 수 있다. 이런 수행 시나리오가 가능한 이유를 설명하라. (4점)

2. List 데이터 타입 (15점)

- (a) 함수 언어를 이용한 프로그램은 List 및 함수 호출로 이루어져 있다. 일반적인 함수 호출은 다음과 같은 prefix 형식이다.

(함수 이름 인수_1 인수_2)

예를 들면 built-in 함수인 '+'를 이용하여 두개의 함수를 더하는 함수 호출은 (+ a b)와 같은 형식으로 표현된다. 이런 함수 언어의 syntax와 아래 member 함수 정의를 참고하여 다음과 같이 입력 두 리스트의 공통 element들의 리스트를 출력하는 함수 intersect를 완성하라. (15점)

```
(DEFINE (member atm a_list)
  (COND
    ((NULL? a_list) #F)
    ((EQ? atm (CAR lis)) #T)
    ((ELSE (member atm (CDR a_list))))
  )
)

(DEFINE (append list1 list2)
  (COND
    ((NULL? list1) list2)
    (ELSE (CONS (CAR list1) (append (CDR list1) list2)))
  )
)

(DEFINE (intersect list1 list2)
  (COND
    ((NULL? list1) '())
    ((member (CAR list1) list2)
      (CONS (CAR list1) (/* (가) 여기를 채워라 (5점) */)
    )
    ( ELSE (/* (나) 여기를 채워라 (5점) */)
  )
)
)
```

수행 예 : (intersect (a b c d) (1 b 4 c)) returns (b c)

- (b) 다음의 Python 프로그램에 의하여 생성되는 List를 써라 (5점)

```
[x * x for x in range(6) if x % 3 == 0]
```

3. 오른쪽과 같은 C-like 언어에서
 (a) static scoping rule을 사용하는 경우
 (b) dynamic scoping rule을 사용하는 경우
 각각 print되는 값은 무엇인가?
 (각 4점 x 2 = 8점)

```
#include <stdio.h>
int aa[5] = {11,22,33,44,55};
main() {
  int bb, *cc;
  cc = &(aa[1]);
  cc += 2;
  bb = *cc++;
}
```

```
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();
  q();
}
```

4. 왼쪽과 같은 C 언어 프로그램에 대하여 다음의 물음에
 답하여라. (8점)

- (a) 이 프로그램 수행이 끝나고 난 후에 aa[]의 내용은 무엇인가? (4점)
 (b) C언어에서 pointer의 type을 하나로 제한하는 이유는 무엇인가? (4점)

5. LISP언어의 list, Associative Array 데이터 타입에 대하여 다음의 빈칸을 채워라.
 (각 3점 x 4 = 12점)

	List (LISP)	Associative Array
data element의 타입	(a)	(b)
data element를 지칭하는 방법	(c)	(d)

6. 다음의 물음에 답하여라. (각 4점 x 3 = 12점)

- (a) Dijkstra의 selection construct나 loop structure는 command앞의 guard는 true이 모든 statement를 nondeterministically 선택하여 수행한다. Ada는 이런 guarded command에 바탕을 둔 control structure를 제공한다. 이런 기능을 제공하는 이유는 무엇인가? 예를 들어 설명하라. (4점)
 (b) Procedure와 Macro의 공통점은 무엇인가? 또한, 이들의 상대적인 장단점을 비교 설명하라. (4점)
 (c) Ada 언어는 parameter passing 방법으로 keyword parameter를 제공한다. 이 방법의 장단점을 설명하라. (4점)

(0=0, 0=0, 1, 1)

7. 아래와 같은 C 프로그램이 32-bit 컴퓨터상에서 수행되었을 때 (a) ~ (d) 문장에 의하여 프린트 되는 값을 써라 (단, word alignment를 한다고 가정한다. print되는 형식에 주의 할 것.) (각 4점 x 4 = 16점)

```
char aaa ;
union utag {
    char b[2]; char *c ; char a[8] ;
} ;
struct aa {
    float j; char aaa; union utag kk ;
} test[10][10], lala;
main() {
    printf("%x %x\n", &aaa, &(test[3][4])); /* 프린트 결과는 4720,4730 */
    test[3][4].kk.c = &aaa ;
    printf("%d\n", sizeof(lala)); /* (a) */
    printf("%x %x \n", &(test[3][4].kk), &(test[3][4].kk)+1); /* (b) */
    printf("%x %x \n", (test[3][4].kk.c), (test[3][4].kk.c)+1); /* (c) */
    printf("%x %x \n", &(test[3][4].kk.c), &(test[3][4].kk.c)+1); /* (d) */
}
```

8. Iterative Statement (각 4점 x 3 = 12점)

- (a) FORTRAN77 언어의 DO 구조에서는 loop 변수의 값이 변하더라도 loop의 제어에는 영향이 없다. 이를 구현하는 방법을 설명하라. (4점)
- (b) 대부분의 언어에서는 Counter-Controlled loop와 Logically Controlled loop 기능을 모두 제공하는 것으로 설계되어 있다. 이런 설계가 orthogonal한 설계인가? 그 이유와 함께 설명하라. (4점)
- (c) Java 언어나 C# 언어에서 제공하는 Iteration based on the Data Structure 기능에 대하여 예를 들어 설명하라. (4점)

9. 다음과 같은 C 언어 프로그램과 Linux (Pentium PC용)상에서 컴파일된 어셈블리 코드를 보고 다음의 질문에 답하라. (각 4점 x 4 = 16점)

- (a) C 언어에서 "=" 와 "+" operator의 associative rule이 어떠한지 어셈블리 코드를 이용하여 설명하라. (4점)
- (b) C 언어에서 "+" operator의 operand evaluation order가 어떠한지 어셈블리 코드를 이용하여 설명하라. (4점)
- (c) coercion이 어떤 방식으로 일어나는지 어셈블리 코드를 이용하여 설명하라. (4점)
- (d) 이 프로그램의 프린트된 결과를 써라 (4점)

```
int a,b,c ; float k ;
int sub(int x) {
    b=b+2;
    return(0); }
main() {
    a=b=c=1; a = a + b + c;
    k = a + b + sub(c) ;
    printf("%f\n", k); }
```

```
.file "test2.c"
.text
.align 4
.globl sub
.type sub,@function
sub:
    pushl %ebp
    movl %esp, %ebp
    addl $2, b
    movl $0, %eax
    popl %ebp
    ret
.Lfe1:
    .size sub, .Lfe1-sub
    .section
.rodata
.LC0:
    .string "%f\n"
```

```
.text
.align 4
.globl main
.type main,@function
main:
    pushl %ebp
    movl %esp, %ebp
    subl $8, %esp
    movl $1, c
    movl $1, b
    movl $1, a
    movl b, %eax
    addl a, %eax
    addl c, %eax
    movl %eax, a
    subl $12, %esp
    pushl c
    call sub
    addl $16, %esp
    movl %eax, %edx
    movl b, %eax
    addl a, %eax
    addl %edx, %eax
    pushl %eax
    fldl (%esp)
    popl %eax
    fstps k
    subl $4, %esp
    flds k
    leal -8(%esp), %esp
    fstpl (%esp)
    pushl $.LC0
    call printf
    addl $16, %esp
    leave
    ret
.Lfe2:
    .size main, .Lfe2-main
    .comm a, 4, 4
    .comm b, 4, 4
    .comm c, 4, 4
    .comm k, 4, 4
    .ident "GCC: (GNU)"
```



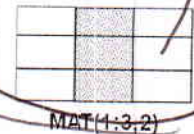
1. 프로그래밍 언어 일반 (8점)

(a) 과학계산 응용에서 많이 사용되는 matrix는 프로그래밍 언어에서 2D array로 표현할 수 있다. 이런 기능은 FORTRAN이나 C 언어에서 모두 표현할 수 있지만, 이 2개의 언어 중에서 FORTRAN을 과학계산용 언어라 할 수 있는 이유를 구체적인 FORTRAN의 기능 예를 들어 설명하라. (4점)

(정답)

- 1) Slice 기능 : 다차원 배열의 일부분 (Slice) 을 지칭하여 계산을 직접 할 수 있는 기능 (예 : Matrix의 일부분을 지칭하여 벡터로 naming할 수 있다.) (2점)

```
INTEGER VECTOR (1:10), MAT (1:3, 1:3), CUBE (1:3, 1:3, 1:3)
MAT = CUBE (1:3, 1:3, 2)
```

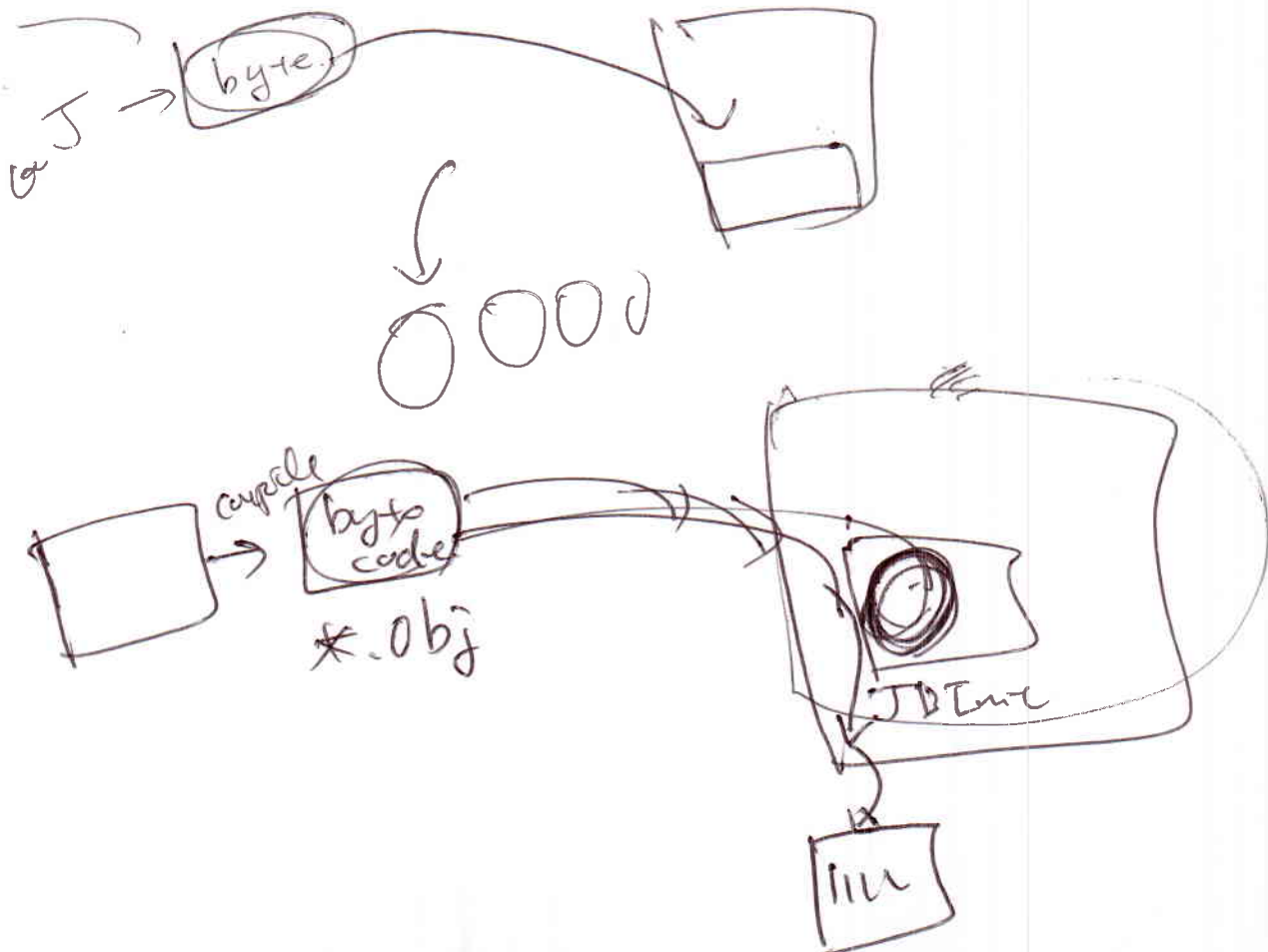


- 2) (다차원) 배열에 대한 사칙연산자의 overloading 기능 (2점)

→ The assignment, arithmetic, relational, and logical operators are overloaded for arrays of any size

(b) Java 언어로 작성된 프로그램은 여러 종류의 platform상에서 Web 브라우저만 있으면 재컴파일할 필요 없이 수행시킬 수 있다. 이런 수행 시나리오가 가능한 이유를 설명하라. (4점)

(정답) Java 언어는 바이트코드 까지만 컴파일되고 이런 바이트 코드를 인터프리팅하는 방식으로 수행된다. 그러므로 JVM이 내장되어 있는 Web 브라우저만 있으면 재컴파일 없이 수행할 수 있다.



2. List 데이터 타입 (15점)

- (a) 함수 언어를 이용한 프로그램은 List 및 함수 호출로 이루어져 있다. 일반적인 함수 호출은 다음과 같은 prefix 형식이다.

(함수_이름 인수_1 인수_2)

예를 들면 built-in 함수인 '+'를 이용하여 두개의 함수를 더하는 함수 호출은 (+ a b)와 같은 형식으로 표현된다. 이런 함수 언어의 syntax와 아래 member 함수 정의를 참고하여 다음과 같이 입력 두 리스트의 공통 element들의 리스트를 출력하는 함수 intersect를 완성하라. (15점)

```
(DEFINE (member atm a_list)
  (COND
    ((NULL? a_list) #F)
    ((EQ? atm (CAR lis)) #T)
    ((ELSE (member atm (CDR a_list))))
  )
)

(DEFINE (append list1 list2)
  (COND
    ((NULL? list1) list2)
    (ELSE (CONS (CAR list1) (append (CDR list1) list2)))
  )
)

(DEFINE (intersect list1 list2)
  (COND
    ((NULL? list1) '())
    ((member (CAR list1) list2)
      (CONS (CAR list1) (/* (가) 여기를 채워라 (5점) */)))
    ( ELSE (/* (나) 여기를 채워라 (5점) */ ) )
  )
)
```

수행 예 : (intersect (a b c d) (1 b 4 c)) returns (b c)

- 가) (intersect (CDR list1) list2) (5점)
 나) (intersect (CDR list1) list2) (5점)

- (b) 다음의 Python 프로그램에 의하여 생성되는 List를 써라 (5점)

```
[x * x for x in range(6) if x % 3 == 0]
```

(정답) [0, 9, 36] (5점)

(풀이 과정) range(6) creates [0, 1, 2, 3, 4, 5, 6] → Constructed list: [0, 9, 36]

3. 오른쪽과 같은 C-like 언어에서
 (a) static scoping rule을 사용하는 경우
 (b) dynamic scoping rule을 사용하는 경우
 각각 print되는 값은 무엇인가?
 (각 4점 x 2 = 8점)

(정답)

- (a) static scoping rule : a=3, b=1
 (b) dynamic scoping rule : a=3, b=4

```
#include <stdio.h>
int aa[5] = {11,22,33,44,55};
main() {
  int bb, *cc;
  cc = &(aa[1]);
  cc += 2;
  bb = *cc++;
}
```

```
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();
  q();
}
```

4. 왼쪽과 같은 C 언어 프로그램에 대하여 다음의 물음에
 답하여라. (각 4점 x 2 = 8점)

- (a) 이 프로그램 수행이 끝나고 난 후에 aa[]의 내용은 무엇인가? (4점)
 (b) C언어에서 pointer의 type을 하나로 제한하는 이유는 무엇인가? (4점)

(정답)

- (a) {11, 22, 33, 45, 55} (4점)
 (b) Type checking에 의한 program의 reliability를 높이기 위하여 (4점)

5. List, Associative Array, Heterogeneous Array, Record 데이터 타입에 대하여
 다음의 빈칸을 채워라. (각 3점 x 4 = 12점)

	List (LISP)	Associative Array
저장되는 data element의 타입	(a) 다른 종류의 데이터 가능	(b) 같은 종류의 데이터
저장되는 data element를 지칭하는 방법	(c) CAR 함수에 의하여 첫 번째 element 지칭, CDR 함수에 의하여 리스트의 tail 부분 지칭	(d) Key

6. 다음의 물음에 답하여라. (각 4점 x 3 = 9점)

(a) Dijkstra의 selection construct나 loop structure는 command 앞의 guard는 true이 모든 statement를 nondeterministically 선택하여 수행한다. Ada는 이런 guarded command에 바탕을 둔 control structure를 제공한다. 이런 기능을 제공하는 이유는 무엇인가? 예를 들어 설명하라. (4점)

(정답) 예를 들어, 여러 곳에서 입력을 받는 프로그램에서 각 입력 장치에 대하여 우선 순위를 두고 싶지 않고자 하는 아이디어를 표현할 수 있게 하기 위하여. 즉, 같이 입력이 들어오면 어떤 것을 입력 받는지 상관 없는 아이디어를 표현할 수 있다. (3점)

(b) Procedure와 Macro의 공통점은 무엇인가? 또한, 이들의 상대적인 장단점을 비교 설명하라. (4점)

(정답)

공통점: Sub프로그램을 정의할 수 있도록 한다.

Procedure 장점: code 사이즈가 작다.

Macro: 프로그램 수행 속도가 빠르다.

for () { }

(c) Ada 언어는 parameter passing 방법으로 keyword parameter를 제공한다. 이 방법의 장단점을 설명하라. (4점)

(정답) 장점: 파라미터를 넘길 때 순서를 지킬 필요가 없으며, 원하는 파라미터만 지정하여 넘길 수 있다.

단점: 프로그램가 formal parameter의 모든 이름을 알아야 한다.

7. 아래와 같은 C 프로그램이 32-bit 컴퓨터상에서 수행되었을 때 (a) ~ (d) 문장에 의하여 프린트 되는 값을 써라 (단, word alignment를 한다고 가정한다. print되는 형식에 주의 할 것.) (각 4점 x 4 = 16점)

```
char aaa ;
union utag {
    char b[2]; char *c ; char a[8] ;
} ;
struct aa {
    float j; char aaa; union utag kk ;
} test[10][10], lala;
main() {
    printf("%x %x\n", &aaa, &(test[3][4])); /* 프린트 결과는 4720,4730 */
    test[3][4].kk.c = &aaa ;
    printf("%d\n", sizeof(lala)); /* (a) */
    printf("%x %x \n", &(test[3][4].kk), &(test[3][4].kk)+1); /* (b) */
    printf("%x %x \n", (test[3][4].kk.c), (test[3][4].kk.c)+1); /* (c) */
    printf("%x %x \n", &(test[3][4].kk.c), &(test[3][4].kk.c)+1); /* (d) */
}
```

(정답) (a) 16

(b) 4738 4740

(c) 4720 4721

(d) 4738 473c

8. Iterative Statement (각 4점 x 3 = 12점)

(a) FORTRAN77 언어의 DO 구조에서는 loop 변수의 값이 변화더라도 loop의 제어에는 영향이 없다. 이를 구현하는 방법을 설명하라. (4점)

(정답) 컴파일러가 loop의 counter를 저장하는 내부 변수 (internal variable)을 생성하고, 이를 runtime시에 1씩 감소하여 횟수를 제어한다. 내부 변수는 프로그램상에서 access할 수 없기 때문에 loop변수가 변화더라도 loop의 횟수는 변하지 않는다.

(b) 대부분의 언어에서는 Counter-Controlled loop와 Logically Controlled loop 기능을 모두 제공하는 것으로 설계되어 있다. 이런 설계가 orthogonal한 설계인가? 그 이유와 함께 설명하라. (4점)

(정답) orthogonal한 설계가 아니다. 그 이유는 기능이 중복되기 때문이다. 즉, logically controlled loop를 이용하여 counter-controlled loop의 기능을 대신할 수 있기 때문이다.

(c) Java 언어나 C# 언어에서 제공하는 Iteration based on the Data Structure 기능에 대하여 예를 들어 설명하라. (4점)

(정답)

loop의 제어가 array나 collection등의 데이터 구조에 저장되어 있는 element 개수에 의하여 제어되기 때문이다.

```
String[] StrList={"Bob", "Carol", "Ted", "lala"};
...
foreach (String name in strList) ... ;
```



9. 다음과 같은 C 언어 프로그램과 Linux (Pentium PC용)상에서 컴파일된 어셈블리 코드를 보고 다음의 물음에 답하라. (각 4점 x 4 = 16점)

(a) C 언어에서 "=" 와 "+" operator의 associative rule이 어떠한지 어셈블리 코드를 이용하여 설명하라. (4점)

(정답)

1) assign된 순서가 c, b, a 이기 때문에 right associative 하다.

$a=b=c=1$

```
movl    $1, c
movl    $1, b
movl    $1, a
```

2) 더하기한 순서가 (a+b)+c 이기 때문에 left associative 하다.

$a = a + b + c$

```
movl    b, %eax
addl    a, %eax
addl    c, %eax
```

(b) C 언어에서 "+" operator의 operand evaluation order가 어떠한지 어셈블리 코드를 이용하여 설명하라. (4점)

(정답) 기본적으로 오른쪽 operand부터 evaluation되지만, 이 코드를 통하여서는 알 수 없다. 반면, 함수 호출이 있는 경우에는 이를 먼저 수행한 후에 '+' operation을 수행한다.

```
movl    b, %eax
addl    a, %eax
addl    c, %eax
```

```
pushl    c
call     sub
addl     $16, %esp
movl     %eax, %edx
movl     b, %eax
addl     a, %eax
addl     %edx, %eax
```

(c) coercion이 어떤 방식으로 일어나는지 어셈블리 코드를 이용하여 설명하라. (4점)

(정답)

$k = a + b + \text{sub}(c)$

```
pushl    %eax    /* a+b+sub(c)를 계산한 결과를 stack에 넣고
fildl    (%esp)  /* 그것을 다시 FPU의 stack에 넣고, convert integer to floating
popl     %eax
fstps    k        /* FPU stack의 top element를 k에 저장시킨다
```

(d) 이 프로그램의 프린트된 결과를 써라 (4점)

(정답) 6.0