**Test #10. 2012.5.14 프로그래밍언어론 00 분반- 개인 문제**

학번:

이름:

조번호:

1. 다음 각 항목이 참이면 O, 아니면 X 를 하시오.
   1. Out mode 또는 Inout mode의 parameter 값은 Subprocedure call 시점에서 actual parameter에 전달된다.
   2. C는 ARI내에 static link 가 필요하지 않다.
   3. 한 subprogram에 대해서 한 시점에 여러 개의 ARI가 존재할 수 있다.
   4. caller 로의 return address(복귀주소) 는 caller나 callee의 ARI에 포함되지 않는다.
   5. chain offset이란 변수를 참조한 block의 static depth와 참조한 변수를 선언한 block의 static depth의 차 (difference)이다.
   6. static chain을 사용하여 static scoping rule을 따르는 변수들을 접근하는 경우 호출 depth가 늘어나면 증가한다.
   7. Static chain을 사용하여 static scoping rule을 따르는 변수들을 접근하는 경우 caller는 callee의 static parent 를 static ancestor로 가진다고 가정한다.
   8. 실행중인 subprogram의 static depth가 n인 경우 display에는 n+1개의 값이 유효하다. (단 depth는 0부터 시작이다.)
   9. ARI의 꼭대기에 block 지역변수가 사용할 기억장소를 할당하는 방법을 쓴다면, 다음 subprogram에서 Block 지역변수가 사용할 기억장소의 최대 크기는 5\*4byte이다.

sub(){

while(1) { int a; int b;}

{int c; …}

{ind d; …}

{int e; …}

}

* 1. 다음 코드에서 g()가 h()를 호출 할 때, display 관리를 위한 아래 i~v 스텝 중에서 ii와 v 중 적어도 하나는 불필요한 과정이다.

main(){

h(){…}

g(){ … h(); …}

g();

}

* + 1. P의 ARI(P-ARI)를 생성
    2. P-ARI 안에 display[k]를 저장
    3. Display[k]가 P-ARI를 point하도록 함
    4. Subprogram P 실행
    5. P-ARI에 저장해둔 이전의 display[k] 값을 display[k]에 저장

2. 다음 그림은 아래 프로그램을 수행했을 때 runtime stack의 일부이다. 그림을 보충하여 stack의 크기가 최대가 되었을 때의 모습으로 완성하시오.

int fact(int n) {

if (n == 0)

return 1;

else

return n\*fact(n-1);

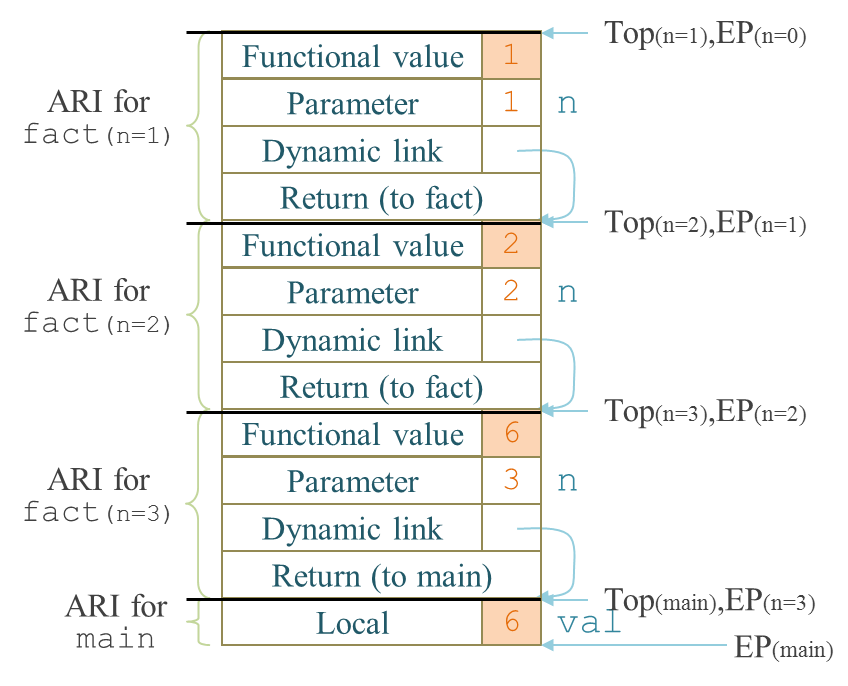
}

void main() {

int val;

val = fact(3);

}



3. 다음 프로그램에서 정의된 sub3()의 수행 중 사용된 E, B, A 의 chain offset 을 각각구하시오.

Main

var X;

proc Bigsub();

var A, B, C;

proc Sub1()

var A, D;

A := B + C;

end;

proc Sub2(X)

var B, E;

proc Sub3()

var C, E;

Sub1();

**E := B + A;**

end;

Sub3();

A := D + E;

end;

Sub2(7);

end;

X := 9;

Bigsub();

end.

**Test #10. 2012.5.14 프로그래밍언어론 00 분반 조별 문제**

조번호:

참여자 (학번, 이름)

-

-

-

-

-

1. 연상

2. 다음 프로그램이 dynamic access rule의 shallow access 방식으로 접근된다면, 함수 sub2()이 처음으로 호출되었을 때 symbol table의 모습을 그려보시오.

void sub3() { int x, z; sub2();...}

void sub2() { int x; x = y + w + z; ......}

void sub1() { int z, w; ... sub3() ...; }

void main() { int x, y; ... sub1(); ... }