4 Runtime Environment

1. In programming language terminology call by value refers to the fact that
(A) A function call can return a value.
(B) When a function is called, arguments are copied into local storage.
(C) Functions can indirectly modify the value of external variables.
(D) Every argument passed to a function must have a value.

2. A linker is given object modules for a set of programs that were compiled separately. What information need to be included in an object module?
(A) Object code
(B)Relocation bits
(C) Names and locations of all external symbols defined in the object module
(D) Absolute addresses of internal symbols

3. Consider following statements about the memory layout of a program in a modern programming language:
(i) A stack is used for storing static variables.
(ii) The size allocated for stack is more than that for heap.
(iii) Malloc() takes a chunk of bytes from heap space and assigns it to a pointer which is stored in heap itself.
(iv) Calloc() takes a chunk of bytes from heap space and assigns it to a pointer which is stored in stack space.
Then which of the above statements is/are not TRUE?
(A) (i) only
(B) (i) and (ii) only
(C) (i), (ii) and (iii)only
(D) All of the above

4. Consider a program P that consists of two source modules A and B contained in two different files. If B contains a reference to a function defined in A the reference will be resolved at:
(A) Edit time
(B) Compile time
(C) Link time
(D) Load time

5. A language L allows declaration of arrays whose sizes are not known during compilation. It is required to make efficient use of memory. Which one of the following is true?
(A) A compiler using static memory allocation can be written for L.
(B) A compiler cannot be written for L an interpreter must be used

(C) A compiler using dynamic memory allocation can be written for L.
(D) None of the above

6. Faster access to non-local variables is achieved using an array of pointers to activation records called a:
(A) stack
(B) heap
(C) display
(D) activation tree

7. Which one of the following is NOT decided during compilation?
(A) Allocating memory to variables.
(B) Type of variables
(C) Symbol table management
(D) Size of array whose size is a function of two variables whose values are in symbol table.
(E) Depth of Call stack

8. A "link editor" is a program that:
(A) matches the parameters of the macro-definition with locations of the parameters of the macro call.
(B) matches external names of one program with their location in other programs.
(C) matches the parameters of subroutine definition with the location of parameters of subroutine call.
(D) acts as a link between text editor and the user.
(E) acts as a link between compiler and the user program.

9. Consider following statements about the statically linked libraries (SLL) and shared dynamically linked libraries (DLL)?
(i) Smaller sizes of executable files in SLL.
(ii) Lesser overall page fault rate in the system in case of DLL.
(iii) Faster program startup in case of SLL.
(iv) Existing programs need not be re-linked to take advantage of newer versions of libraries in case of SLL
Which of the above is/are correct?
(A) (i) only
(B) (iii) only
(C) (i) and (iv) only
(D) (ii) and (iii)only
(E) (i), (ii) and (iii)only
(F) All of the above

10. Which of the following statements are CORRECT?
(i) Static allocation of all data areas by a compiler makes it impossible to implement recursion.
(ii) Automatic garbage collection is essential to implement recursion.
(iii) Dynamic allocation of activation records is essential to implement recursion.
(iv) Both heap and stack are essential to implement recursion.
(A) (i) and (ii) only
(B) (ii) and (iii) only
(C) (iii) and (iv) only
(D) (i) and (iii) only

11. Consider the following function.

```
void magic(int *a,int *b)
{
*a+ = *b;
*b- = *a;
*b* = -1;
*a- = *b;
}
```

What is above function doing?
(A) Swaps values of a and b if called as magic(x, y)
(B) Swaps values of a and b if called as magic(x, y)
(C) magic(x, y) can't be used as it does not return any value
(D) None.

12. What are and in the following macro definition?

```
macro Add x, y
Load y
Mul x
Store y
end macro
```

(A) Variables
(B) Identifiers
(C) Actual parameters
(D) Formal parameters

13. Consider following statements:
(i) Recursive descent parsing cannot be used for grammar with left recursion.
(ii) One of the intermediate form for representing expressions used for code optimization is the postfix form.
(iii) A programming language not supporting either recursion or pointer type does not need the support of dynamic memory allocation.

(iv) Although C does not support call-by-name parameter passing, the effect can be correctly simulated in C.

Which of the above statements are correct?
(A) (i) only
(B) (iii) only
(C) (i) and (iv) only
(D) (ii) and (iii)only
(E) (i), (ii) and (iv)only
(F) All of the above

14. The conditional expansion facility of macro processor is provided to
(A) test a condition during the execution of the expanded program.
(B) to expand certain model statements depending upon the value of a condition during the execution of the expanded program.
(C) to implement recursion.
(D) to expand certain model statements depending upon the value of a condition during the process of macro expansion.

15. Consider following statements about scoping rules:
(i) A non-local name “a” in the called activation refers to a global storage and not to that of the calling activation in case of static scoping.
(ii) A non-local name “a” in the called activation refers to same storage that it did in the calling activation in case of dynamic scoping.
(iii) A non-local name “a” in the called activation refers to a global storage and not to that of the calling activation in case of dynamic scoping.
(iv) A non-local name “a” in the called activation refers to same storage that it did in the calling activation in case of static scoping.
Which of the above statements are not necessarily true?
(A) (i) and (ii) only
(B) (iii) only
(C) (i) and (iv) only
(D) (iii) and (iv)only
(E) (i), (ii) and (iv) only
(F) All of the above

16. Consider following statements about deep and shallow access of non-locals in case of dynamic scoping:
(i) Use of access links is avoided and instead control links are used to search into the stack in case of deep access.
(ii) Shallow access allows to hold current value of each name in static memory and when a new activation of P occurs a local name “n” in P takes over the storage for “n”.
(iii) Control links as well as access links are used to search into the stack

in case of deep access in both upward and downward directions.
(iv) Shallow access allows to hold all values of every name in heap based array like structure and when a new activation of P occurs a local name "n" in P is inserted into this array along with nonlocals "n" with a subscript attached to distinguish it from nonlocals.
Which of the above statements are necessarily true?
(A) (i) and (ii) only
(B) (iii) only
(C) (i) and (iv) only
(D) (iii) and (iv)only
(E) (i), (ii) and (iv) only
(F) All of the above

17. What is the value of X printed by the following program?

```
program COMPUTE (input, output);
var X:integer;
procedure FIND (X:real);
begin
X:=sqrt(X)*sqrt(X)*sqrt(X);
end;
begin
X:=4;
FIND(X);
writeln(X);
end
```

(A) $2\sqrt{2}$
(B) $4\sqrt{2}$
(C) 8
(D) 64

18. What will be the output of the following pseudo-code when parameters are passed by reference and static scoping is assumed?

```
int a = 8;
void n(int x)
{
x *= a;
print (x);
}
void m(int y)
{
int a = 4 ;
a = y - a;
n(a);
print (a);
}
```

```
void main ()
{
m(a);
}
```

(A) 12,12
(B) 12,16
(C) 16,16
(D) 16,32
(E) 32,32

19. Consider following pseudo-code:

```
program dynamic (input, output);
        var r: real;
procedure show;
        begin write(r) end;
procedure small;
        var r: real;
        begin r := 0.125; show end;
begin
        r := 0.25;
        show; small; writeln;
        show; small; writeln;
end.
```

What will be the output of the above pseudo-code when :
(i) static scoping is assumed.
(ii) dynamic scoping is assumed.

20. Which of the following information field is not stored as part of an activation record? (More than one can be correct.)
(A) Access link
(B) Control link
(C) Machine status
(D) Temporary variables
(E) Static variables
(F) Local variables
(G) Base address of page table

21. An activation record contains state information for a particular instance of a function call. What are the uses of access links and control links information found in an activation record?
(A) Access link is used to access non local data and control link points to activation record of called function.
(B) Access link is used to hold values to be returned to caller function and control link points to activation record of called function.

(C)Access link is used to hold values to be returned to caller function and control link points to activation record of caller function.
(D) Access link is used to access non local data and control link points to activation record of caller function.

22. Consider the program below in a hypothetical language which allows global variable and a choice of call by reference or call by value methods of parameter passing.

```
int i,j ;
program main ()
{
int j = 60;
i = 50;
call f (i, j);
print i, j;
}
procedure f (x, y)
{
i = 100;
x = x - 5;
y += i + j ;
}
```

Which one of the following options represents the correct output of the program for the two parameter passing mechanisms?
(A) Call by value : i = 45, j = 60; Call by reference : i = 95, j = 160
(B) Call by value : i = 50, j = 60; Call by reference : i = 50, j = 70
(C) Call by value : i = 100, j = 70; Call by reference : i = 100, j = 160
(D) Call by value : i = 100, j = 60; Call by reference : i = 95, j = 215

23. What is printed by following program, assuming call-by reference method of passing parameters for all variables in the parameter list of procedure P?

```
program Main(input, output);
var a, b:integer;
procedure P(x, y, z:integer);
begin
        y:=x+1;
        x+=y+1;
        z:=z+x;
end P;
begin
        a:=5;
        b:=4;
        p(a+b, a, a);
        Write(a);
```

```
end;
```

(A) 5
(B) 9
(C) 15
(D) 25
(E) 29
(F) 30

24. Consider the following pseudo-code (all data items are of type integer):

```
procedure P(a, b, c, d);
a := 2;
c := a + b;
d := 2*c+1;
end P
begin
x := 1;
y := 5;
z := 100;
P(z, x*y, x, y);
Write ('x = ', x, 'y = ',y ,'z = ', z);
end
```

Determine its output, if the parameters are passed to the Procedure P by:
(i) call by value
(ii) call by reference
(iii) call by name
(iv) call by copy-restore
(v) call by need

25. What is the output of the following program if parameters are :
(i)passed by name
(ii) passed by reference
(iii) passed by value

```
program P(output);
    var A: array [1..4] of integer;
    I, J: integer;
    procedure R(X,Y,Z: integer);
    begin
        I := I + 1;
        J := J + 2;
        X := X * Z;
        Y := 0
    end;
    procedure Q;
    var J,K: integer;
```

```
begin
      J := 2;
      K := 3;
      R(A[I], A[J], J + K)
end;
begin
for J := 1 to 4 do A[J] := 3;
J := 1;
I := 2;
Q;
writeln(A[1],A[2],A[3],A[4])
end.
```

26. For each of the parameter passing methods what are the values of the list array after execution?
(i) Pass by value
(ii) Pass by reference
(iii) Pass by value result
(iv) Pass by need
(v) Pass by name

```
#include <stdio.h>
void fun(int a, int b);
void main()
{
int list[3] = {1,2,3};
fun(list[2],list[0],list[1]);
printf(”List contains: ”);
int i;
for(i = 0; i < 3; i++)
        printf(”%d ”,list[i]);
}
void fun(int first, int second)
{
first += first;
second += second;
third += third;
}
```

27. Consider the following code:

```
program p(output);
      var a,b,i: integer;
      c: array [1..4] of integer;
      procedure s(d,e,f: integer);
      begin
```

```
            d := 5; b := a+3; e := 3; i:= 1; f:= 20; c[i] := 200;
    end;
    begin
        c[1] := 100; c[2] := 150; c[3] := 175; c[4] := 190;
        a := 10; b := 5; i := 2;
        s(a,i,c[i]);
        writeln(a,b,c[1],c[2],c[3],c[4]);
    end.
```

For each to the five parameter passing techniques shown below, give the output of the following program.
(i) Pass by value
(ii) Pass by reference
(iii) Pass by value result
(iv) Pass by need
(v) Pass by name

28. We define a new parameter-passing technique, call by lazy evaluation. When a parameter is passed, it is not evaluated until the code of the called procedure actually needs the value. Then evaluation takes place, the result is stored in a temporary location and further references use the stored value. What would the following program print if the language of this program has this parameter passing scheme as its default parameter passing mechanism?

```
program P(output);
    function R(A,B: integer): integer;
    begin
        if B = 0 then R := 0;
    else R := R(A,B);
    end;
    begin
        writeln(R(R(1,1),R(0,0)))
    end.
```

29. Consider following code fragment:

```
int foo(int x)
{
int a;
if(x==0) return 1;
else
{
      a=foo(x-1);
      return (x*a);
}
}
```

How many overall activation records in total will be there in the lifetime of this code fragment if $foo(6)$ is called?
(A) 6
(B) 7
(C) 8
(D) 120
(E) 720

30. Suppose procedure P at depth d_P refers to a non-local “a'' at depth d_a such that $(d_a \leq d_P)$, then storage for a can be found by?
(A) following $(d_P - d_a + 1)$ control links from the record at the top of the stack.
(B) following $(d_a - d_P - 1)$ access links from the record at the top of the stack.
(C) following $(d_P - d_a + 1)$ control links from the record at the top of the stack.
(D) following $(d_P - d_a)$ access links from the record at the top of the stack.

31. Suppose procedure P at depth d_P calls procedure Q at depth d_Q such that $d_P \geq d_Q$. How many links are to be followed to get the definition of the called function?
(A) $(d_P - d_Q - 1)$
(B) $(d_Q - d_P - 1)$
(C) $(d_P - d_Q + 1)$
(D) $(d_P - d_Q + 1)$

32. In the above question number of access links are to be stored inside Symbol table. The number of links can be computed as earliest as at:
(A) Compile time
(B) Run time
(C) Load time
(D) None of the above

33. Consider following program:

```
program P2(output);
    var A: integer;
    procedure S (procedure T(var X: integer)); forward;
    procedure Q (var B: integer);
        var C: integer;
        begin
            C := B + A;
            writeln(C);
            if B > 0 then S(Q)
        end;
    procedure S;
        var C: integer;
        begin
            C := 0;
            T(C)
        end;
    begin
        A := 7;
        Q(A)
    end.
program P3(output);
    var Z: integer;
    procedure T(procedure U);
        begin
        Z := Z - 2;
        U
        end;
    procedure Q(procedure R(procedure S); var X: integer);
        var Y: integer;
        procedure V;
        begin
            X := X + 1
        end;
        begin
            Y := 2;
            X := Z - Y;
            R(V)
        end;
begin
    Z := 6;
    Q(T,Z);
    writeln(Z)
end.
```

For the above program, show the contents of the stack at its highest point using
(a) static links
(b) displays