C Structures Questions

Sayak Haldar
IIEST, Shibpur

1) Which of the following are themselves a collection of different data types?

a) string
b) structures
c) char
d) All of the mentioned

2) User-defined data type can be derived by_____.

a) struct
b) enum
c) typedef
d) All of the mentioned

3) Which operator connects the structure name to its member name?

a) -
b) <-
c) .
d) Both (b) and (c)

4) Which of the following cannot be a structure member?

a) Another structure
b) Function
c) Array
d) None of the mentioned

5) Which of the following structure declaration will throw an error?

a) struct temp{}s;
   main(){}
b) struct temp{};
   struct temp s;
   main(){}
c) struct temp s;
   struct temp{};
   main(){}
d) None of the mentioned

6) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no;
```

```
5.      char name[20];
6.  }
7.  int main()
8.  {
9.      struct student s;
10.     s.no = 8;
11.     printf("hello");
12.     return 0;
13.  }
```

a) Compile time error
b) Nothing
c) hello
d) Varies

7) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no = 5;
5.      char name[20];
6.  };
7.  int main()
8.  {
9.      struct student s;
10.     s.no = 8;
11.     printf("hello");
12.     return 0;
```

```
13.  }
```

a) Nothing
b) Compile time error
c) hello
d) Varies

8) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      int no;
5.      char name[20];
6.  };
7.  int main()
8.  {
9.      student s;
10.     s.no = 8;
11.     printf("hello");
12.     return 0;
13.  }
```

a) Nothing
b) hello
c) Compile time error
d) Varies

9) What is the output of this C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      struct student
```

```
5.      {
6.          int no;
7.          char name[20];
8.      };
9.      struct student s;
10.     s.no = 8;
11.     printf("%d", s.no);
12.     return 0;
13. }
```

a) Nothing
b) Compile time error
c) Junk
d) 8

10) Can the above code be compiled successfully?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int main()
9.  {
10.     struct p x = {.c = 97, .f = 3, .k = 1};
11.     printf("%f\n", x.f);
12.     return 0;
```

```
13. }
```

a) Yes
b) No
c) Depends on the standard
d) Depends on the platform

11) What is the output of this C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      struct student
5.      {
6.          int no;
7.          char name[20];
8.      };
9.      struct student s;
10.     no = 8;
11.     printf("%d", no);
12.     return 0;
13. }
```

a) Nothing
b) Compile time error
c) Junk
d) 8

12) Number of bytes in memory taken by the below structure is

```
1.  struct test
2.  {
3.      int k;
4.      char c;
```

```
5.  };
```

a) Multiple of integer size
b) integer size+character size
c) Depends on the platform
d) Multiple of word size

13) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct
3.  {
4.      int k;
5.      char c;
6.  };
7.  int main()
8.  {
9.      struct p;
10.     p.k = 10;
11.     printf("%d\n", p.k);
12.    return 0;
13. }
```

a) Compile time error
b) 10
c) Undefined behaviour
d) Segmentation fault

14) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct
3.  {
4.      int k;
```

```
5.      char c;
6.  } p;
7.  int p = 10;
8.  int main()
9.  {
10.     p.k = 10;
11.     printf("%d %d\n", p.k, p);
12.      return 0;
13.  }
```

a) Compile time error
b) 10 10
c) Depends on the standard
d) Depends on the compiler

15) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.  };
7.  int p = 10;
8.  int main()
9.  {
10.     struct p x;
11.     x.k = 10;
12.     printf("%d %d\n", x.k, p);
```

```
13.     return 0;
14. }
```

a) Compile time error
b) 10 10
c) Depends on the standard
d) Depends on the compiler

16) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int p = 10;
9.  int main()
10. {
11.     struct p x = {1, 97};
12.     printf("%d %c %f %d\n",x.k,x.c,x.f, p);
13.     return 0;
14. }
```

a) Compile time error
b)1 a 0.000000 10
c) Somegarbage value Somegarbage value Somegarbage value 10
d) 0 \0 0 10

17) What is the output of this C code (according to C99 standard)?

```
1.  #include <stdio.h>
2.  struct p
```

```
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int main()
9.  {
10.     struct p x = {.c = 97, .f = 3, .k = 1};
11.     printf("%f\n", x.f);
12.     return 0;
13. }
```

a) 3.000000
b) Compile time error
c) Undefined behaviour
d) 1.000000

18) What is the output of this C code(according to C99 standard)?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int main()
9.  {
10.     struct p x = {.c = 97, .k = 1, 3};
```

```
11.     printf("%f \n", x.f);
12.     return 0;
13. }
```

a) 3.000000
b) 0.000000
c) Compile time error
d) Undefined behaviour

19) What is the output of this C code(according to C99 standard)?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int k;
5.      char c;
6.      float f;
7.  };
8.  int main()
9.  {
10.     struct p x = {.c = 97};
11.     printf("%f\n", x.f);
12.     return 0;
13. }
```

a) 0.000000
b) Somegarbagevalue
c) Compile time error
d) None of the mentioned

20) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
```

```
3.  {
4.      char *name;
5.  };
6.  struct student s;
7.  struct student fun(void)
8.  {
9.      s.name = "newton";
10.     printf("%s\n", s.name);
11.     s.name = "alan";
12.     return s;
13. }
14. int main()
15. {
16.     struct student m = fun();
17.     printf("%s\n", m.name);
18.     m.name = "turing";
19.     printf("%s\n", s.name);
20.     return 0;
21. }
```

a) newton alan alan
b) alan newton alan
c) alan alan newton
d) Compile time error

21) The correct syntax to access the member of the ith structure in the array of structures is?

Assuming: struct temp

```
{
    int b;
```

}s[50];
a) s.b.[i];
b) s.[i].b;
c) s.b[i];
d) s[i].b;

22) Comment on the output of this C code?

```
1.  #include <stdio.h>
2.  struct temp
3.  {
4.      int a;
5.      int b;
6.      int c;
7.  };
8.  int main()
9.  {
10.     struct temp p[] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
11.     return 0;
12. }
```

a) No Compile time error, generates an array of structure of size 3
b) No Compile time error, generates an array of structure of size 9
c) Compile time error, illegal declaration of a multidimensional array
d) Compile time error, illegal assignment to members of structure

23) Which of the following uses structure?

a) Array of structures
b) Linked Lists
c) Binary Tree
d) All of the mentioned

24) What is the correct syntax to declare a function foo () which receives an array of structure in function?

a) void foo(struct *var);
b) void foo(struct *var[]);

c) void foo(struct var);
d) None of the mentioned

25) What is the output of this C code?

(Assuming size of int be 4)

```
1.  #include <stdio.h>
2.  struct temp
3.  {
4.      int a;
5.      int b;
6.      int c;
7.  } p[] = {0};
8.  int main()
9.  {
10.     printf("%d", sizeof(p));
11.     return 0;
12. }
```

a) 4
b) 12
c) 16
d) Can’t be estimated due to ambiguous initialization of array

26) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  struct student s[2];
7.  int main()
```

```
8.  {
9.      s[0].name = "alan";
10.     s[1] = s[0];
11.     printf("%s%s", s[0].name, s[1].name);
12.     s[1].name = "turing";
13.     printf("%s%s", s[0].name, s[1].name);
14.     return 0;
15. }
```

a) alan alan alan turing
b) alan alan turing turing
c) alan turing alan turing
d) Run time error

27) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  struct student s[2], r[2];
7.  void main()
8.  {
9.      s[0].name = "alan";
10.     s[1] = s[0];
11.     r = s;
12.     printf("%s%s", r[0].name, r[1].name);
13. }
```

a) alan alan
b) Compile time error
c) Varies
d) Nothing

28) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *name;
5.  };
6.  int main()
7.  {
8.      struct student s[2], r[2];
9.      s[1] = s[0] = "alan";
10.     printf("%s%s", s[0].name, s[1].name);
11.     return 0;
12. }
```

a) alan alan
b) Nothing
c) Compile time error
d) Varies

29) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.  };
5.  int main()
6.  {
```

```
7.      struct student s[2];
8.      printf("%d", sizeof(s));
9.      return 0.
10.  }
```

a) 2
b) 4
c) 8
d) 0

30) What is the output of this C code?

```
1.   #include <stdio.h>
2.   struct point
3.   {
4.      int x;
5.      int y;
6.   };
7.   int main()
8.   {
9.      struct point p = {1};
10.     struct point p1 = {1};
11.     if(p == p1)
12.        printf("equal\n");
13.     else
14.        printf("not equal\n");
15.    return 0;
16.  }
```

a) Compile time error
b) equal

c) depends on the standard
d) not equal

31) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  struct notpoint
8.  {
9.      int x;
10.     int y;
11. };
12. struct point foo();
13. int main()
14. {
15.     struct point p = {1};
16.     struct notpoint p1 = {2, 3};
17.     p1 = foo();
18.     printf("%d\n", p1.x);
19. }
20. struct point foo()
21. {
22.     struct point temp = {1, 2};
23.     return temp;
```

```
24.  }
```

a) Compile time error
b) 1
c) 2
d) Undefined behaviour

32) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  struct notpoint
8.  {
9.      int x;
10.     int y;
11. };
12. int main()
13. {
14.     struct point p = {1};
15.     struct notpoint p1 = p;
16.     printf("%d\n", p1.x);
17.    return 0;.
18. }
```

a) Compile time error
b) 1
c) 0
d) Undefined

33) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  struct notpoint
8.  {
9.      int x;
10.     int y;
11. };
12. void foo(struct point);
13. int main()
14. {
15.     struct notpoint p1 = {1, 2};
16.     foo(p1);
17.     return 0;
18. }
19. void foo(struct point p)
20. {
21.     printf("%d\n", p.x);
22. }
```

a) Compile time error
b) 1
c) 0

d) Undefined

34) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1 = {1, 2};
11.     foo(&p1);

12.      return 0;
13.  }
14.  void foo(struct point *p)
15.  {
16.     printf("%d\n", *p.x++);
17.  }
```

a) Compile time error
b) Segmentation fault/code crash
c) 2
d) 1

35) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
```

```
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1 = {1, 2};
11.     foo(&p1);
12.     return 0;
13. }
14. void foo(struct point *p)
15. {
16.     printf("%d\n", *p->x++);
17. }
```

a) Compile time error
b) 1
c) Segmentation fault/code crash
d) 2

36) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student fun(void)
3.  {
4.      struct student
5.      {
6.          char *name;
```

```
7.      };
8.      struct student s;
9.      s.name = "alan";
10.     return s;
11.  }
12.  void main()
13.  {
14.     struct student m = fun();
15.     printf("%s", m.name);
16.  }
```

a) Compile time error
b) alan
c) Nothing
d) Varies

37) What is the output of this C code?

```
1.   #include <stdio.h>
2.   struct point
3.   {
4.      int x;
5.      int y;
6.   };
7.   void foo(struct point*);
8.   int main()
9.   {
10.     struct point p1[]  =  {1, 2, 3, 4};
11.     foo(p1);
```

```
12. }
13. void foo(struct point p[])
14. {
15.     printf("%d\n", p[1].x);
16. }
```

a) Compile time error
b) 3
c) 2
d) 1

37) What is the output of this C code?

```
1. #include <stdio.h>
2. struct point
3. {
4.     int x;
5.     int y;
6. };
7. void foo(struct point*);
8. int main()
9. {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12. }
13. void foo(struct point p[])
14. {
15.     printf("%d\n", p->x);
16. }
```

a) 1
b) 2
c) 3
d) Compile time error

38) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->x, ++p->x);
17. }
```

a) 1 2
b) 2 2
c) Compile time error
d) Undefined behaviour

39) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d\n",++p->x);
17. }
```

a) 1
b) 2
c) Compile time error
d) Undefined behaviour

40) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
```

```
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d\n", p->x);
17. }
```

a) 1
b) 2
c) Compile time error
d) Undefined behaviour

41) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
```

```
9.  {
10.     struct point p1[] = {1, 2, 3, 4};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->y, ++p->y);
17. }
```

a) 2 3
b) 3 3
c) Compile time error
d) Undefined behaviour

42) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.     int x;
5.     int y;
6.  } p[] = {1, 2, 3, 4, 5};
7.  void foo(struct point*);
8.  int main()
9.  {
10.     foo(p);
11.     return 0;
12. }
```

```
13. void foo(struct point p[])
14. {
15.     printf("%d %d\n", p->x, p[2].y);
16. }
```

a) 1 0
b) Compile time error
c) 1 somegarbagevalue
d) Undefined behaviour

43) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p[] = {1, 2, 3, 4, 5};
11.     foo(p);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->x, p[2].y);
17. }
```

a) 1 0
b) Compile time error
c) 1 somegarbagevalue
d) Undefined behaviour

44) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4, 5};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->x, p[3].y);
17. }
```

a) Compile time error
b) 1 0
c) 1 somegarbagevalue
d) None of the mentioned

45) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4, 5};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->x, (p + 2).y);
17. }
```

a) Compile time error
b) 1 0
c) 1 somegarbagevalue
d) Undefined behaviour

46) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct point
3.  {
4.      int x;
```

```
5.      int y;
6.  };
7.  void foo(struct point*);
8.  int main()
9.  {
10.     struct point p1[] = {1, 2, 3, 4, 5};
11.     foo(p1);
12.     return 0;
13. }
14. void foo(struct point p[])
15. {
16.     printf("%d %d\n", p->x, (p + 2)->y);
17. }
```

a) Compile time error
b) 1 0
c) 1 somegarbagevalue
d) undefined behaviour

47) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.  };
6.  int main()
7.  {
8.      struct student s[2];
```

```
9.      printf("%d", sizeof(s));
10.     return 0;
11. }
```

a) 2
b) 4
c) 16
d) 8

48) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      char y;
6.  };
7.  int main()
8.  {
9.      struct p p1[] = {1, 92, 3, 94, 5, 96};
10.     struct p *ptr1 = p1;
11.     int x = (sizeof(p1) / 3);
12.     if (x == sizeof(int) + sizeof(char))
13.         printf("%d\n", ptr1->x);
14.     else
15.         printf("false\n");
16.     return 0;
17. }
```

a) Compile time error
b) 1

c) Undefined behaviour
d) false

49) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      char y;
6.  };
7.  typedef struct p* q*;
8.  int main()
9.  {
10.     struct p p1[] = {1, 92, 3, 94, 5, 96};
11.     q ptr1 = p1;
12.     printf("%d\n", ptr1->x);
13.      return 0;
14. }
```

a) Compile time error
b) 1
c) Undefined behaviour
d) Segmentation fault

50) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      char y;
```

```
6.  };
7.  void foo(struct p* );
8.  int main()
9.  {
10.     typedef struct p* q;
11.     struct p p1[] = {1, 92, 3, 94, 5, 96};
12.     foo(p1);
13.     return 0;
14. }
15. void foo(struct p* p1)
16. {
17.     q ptr1 = p1;
18.     printf("%d\n", ptr1->x);
19. }
```

a) Compile time error
b) 1
c) Segmentation fault
d) Undefined behaviour

51) Which of the following are incorrect syntax for pointer to structure?

(Assuming struct temp{int b;}*my_struct;)
a) *my_struct.b = 10;
b) (*my_struct).b = 10;
c) my_struct->b = 10;
d) Both (a) and (b)

52) For the following function call which option is not possible? func(&s.a); //where s is a variable of type struct and a is the member of the struct.

a) Compiler can access entire structure from the function.
b) Individual member's address can be displayed in structure.
c) Individual member can be passed by reference in a function.
d) Both (b) and (c).

53) Which of the following is an incorrect syntax to pass by reference a member of a structure in a function?
(Assume: struct temp{int a;}s;)

a) func(&s.a);
b) func(&(s).a);
c) func(&(s.a));
d) None of the mentioned

54) Comment on the output of this C code?

```
1.  #include <stdio.h>
2.  struct temp
3.  {
4.      int a;
5.  } s;
6.  void change(struct temp);
7.  int  main()
8.  {
9.      s.a = 10;
10.     change(s);
11.     printf("%d\n", s.a);
12.    return 0;
13.  }
14.  void change(struct temp s)
15.  {
16.     s.a = 1;
17.  }
```

a) Output will be 1
b) Output will be 10
c) Output varies with machine

d) Compile time error

55) What is the output of this C code?

```
1. #include <stdio.h>
2.  struct p
3.  {
4.     int x;
5.     int y;
6.  };
7.  int main()
8.  {
9.     struct p p1[] = {1, 92, 3, 94, 5, 96};
10.    struct p *ptr1 = p1;
11.    int x = (sizeof(p1) / 5);
12.    if (x == 3)
13.        printf("%d %d\n", ptr1->x, (ptr1 + x - 1)->x);
14.    else
15.        printf("false\n");
16. }
```

a) Compile time error
b) 1 5
c) Undefined behaviour
d) false

56) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
```

```
4.      char *c;
5.      struct student *point;
6.  };
7.  int main()
8.  {
9.      struct student s;
10.     struct student m;
11.     s.c = m.c = "hi";
12.     m.point = &s;
13.     (m.point)->c = "hey";
14.     printf("%s\t%s\t", s.c, m.c);
15.     return 0;
16. }
```

a) hey hi
b) hi hey
c) Run time error
d) hey hey

57) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.      struct student *point;
6.  };
7.  int main()
8.  {
```

```
9.      struct student s;
10.     struct student m;
11.     m.point = s;
12.     (m.point)->c = "hey";
13.     printf("%s", s.c);
14.     return 0;
15. }
```

a) Nothing
b) Compile time error
c) hey
d) Varies

58) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.      struct student point;
6.  };
7.  int main()
8.  {
9.      struct student s;
10.     s.c = "hello";
11.     printf("%s", s.c);
12.     return 0;
13. }
```

a) hello
b) Nothing

c) Varies
d) Compile time error

59) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.      struct student *point;
6.  };
7.  int main()
8.  {
9.      struct student s;
10.     printf("%d", sizeof(s));
11.     return 0;
12. }
```

a) 5
b) 9
c) 8
d) 16

60) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char *c;
5.      struct student *point;
6.  };
7.  int main()
```

```
8.  {
9.      struct student s;
10.     struct student *m = &s;
11.     printf("%d", sizeof(student));
12.     return 0;
13. }
```

a) Compile time error
b) 8
c) 5
d) 16

61) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      int x;
5.      char y;
6.      struct p *ptr;
7.  };
8.  int main()
9.  {
10.     struct p p = {1, 2, &p};
11.     printf("%d\n", p.ptr->x);
12.     return 0;
13. }
```

a) Compile time error
b) Undefined behaviour
c) 1
d) 2

62) What is the output of this C code?

```
1.  #include <stdio.h>
2.  typedef struct p *q;
3.  struct p
4.  {
5.      int x;
6.      char y;
7.      q ptr;
8.  };
9.  typedef struct p *q;
10. int main()
11. {
12.     struct p p = {1, 2, &p};
13.     printf("%d\n", p.ptr->x);
14.     return 0;
15. }
```

a) Compile time error
b) 1
c) Undefined behaviour
d) Address of p

63) What is the output of this C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      typedef struct p *q;
5.      struct p
```

```
6.      {
7.          int x;
8.          char y;
9.          q ptr;
10.     };
11.     struct p p = {1, 2, &p};
12.     printf("%d\n", p.ptr->x);
13.     return 0;
14. }
```

a) Compile time error
b) 1
c) Depends on the compiler
d) Depends on the standard

64) What is the output of this C code?

```
1.  #include <stdio.h>
2.  typedef struct p *q;
3.  struct p
4.  {
5.      int x;
6.      char y;
7.      q ptr;
8.  };
9.  int main()
10. {
11.     struct p p = {1, 2, &p};
12.     printf("%d\n", p.ptr->ptr->x);
```

```
13.     return 0;
14. }
```

a) Compile time error
b) Segmentation fault
c) Undefined behaviour
d) 1

65) The number of distinct nodes the following struct declaration can point to is.

```
1.  struct node
2.  {
3.      struct node *left;
4.      struct node *centre;
5.      struct node *right;
6.  };
```

a) 1
b) 2
c) 3
d) All of the mentioned

66) Which of the following is not possible?

a) A structure variable pointing to itself
b) A structure variable pointing to another structure variable of same type
c) 2 different type of structure variable pointing at each other.
d) None of these

67) Which of the following technique is faster for traveling in binary trees?

a) Iteration
b) Recursion
c) Both (a) and (b)
d) Depends from compiler to compiler

68) For the following declaration of structure, which of the following will stop the loop at the last node of a linked list?

```
1.  struct node
2.  {
3.      struct node *next;
4.  };
```

a)
```
while (p != NULL)
  {
      p = p->next;
  }
```
b)
```
while (p->next != NULL)
  {
      p = p->next;
  }
```
c)
```
while (1)
  {
      p = p->next;
      if (p == NULL)
          break;
  }
```
d) All of the mentioned

69)What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct student
3.  {
4.      char a[5];
5.  };
6.  int main()
7.  {
8.      struct student s[] = {"hi", "hey"};
9.      printf("%c", s[0].a[1]);
10.     return 0;
11. }
```

a) h
b) i
c) e
d) y

70) What is the output of this C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      char *a[3] = {"hello", "this"};
5.      printf("%s", a[1]);
6.      return 0;
7.  }
```

a) hello
b) Varies
c) this
d) Compile time error

71) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      char *name;
5.      struct p *next;
6.  };
7.  struct p *ptrary[10];
8.  int main()
9.  {
10.     struct p p;
```

```
11.     p->name = "xyz";
12.     p->next = NULL;
13.     ptrary[0] = &p;
14.     printf("%s\n", p->name);
15.     return 0;
16. }
```

a) Compile time error
b) Segmentation fault/code crash
c) xyz
d) Undefined behaviour

72) What is the output of this C code?

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      char *name;
5.      struct p *next;
6.  };
7.  struct p *ptrary[10];
8.  int main()
9.  {
10.     struct p p;
11.     p.name = "xyz";
12.     p.next = NULL;
13.     ptrary[0] = &p;
14.     printf("%s\n", ptrary[0]->name);
15.     return 0;
```

```
16. }
```

a) Compile time error
b) Segmentation fault
c) Undefined behaviour
d) xyz

73)**What is the output of this C code?**

```
1.  #include <stdio.h>
2.  struct p
3.  {
4.      char *name;
5.      struct p *next;
6.  };
7.  struct p *ptrary[10];
8.  int main()
9.  {
10.     struct p p, q;
11.     p.name = "xyz";
12.     p.next = NULL;
13.     ptrary[0] = &p;
14.     strcpy(q.name, p.name);
15.     ptrary[1] = &q;
16.     printf("%s\n", ptrary[1]->name);
17.     return 0;
18. }
```

a) Compile time error
b) Segmentation fault/code crash
c) Depends on the compiler
d) xyz

74) Which function is responsible searching in the table?
(For #define IN 1, the name IN and replacement text 1 are stored in a “table”)

a) findout(s);
b) lookup(s);
c) find(s);
d) lookfor(s);

75) Which algorithm is used for searching in the table?
a) List search
b) Informed search
c) Hash search
d) Adversarial search

76) Which function is responsible for recording the name “s” and the replacement text “t” in a table?
a) install(s, t);
b) fix(s, t);
c) setup(s, t);
d) settle(s, t);

77) Which of the following is true?

a) Install function uses lookup
b) lookup function uses install
c) Install and lookup function work independently
d) Both (a) as well as (b)

78) What happens when install(s, t) finds that the name being installed is already present in table?
a) It doesn’t modify the name in the table
b) It modifies the name with new definition
c) It modifies if the new definition has higher priority
d) It creates a new table and add the new definition in it

79) In what situation, install function returns NULL?

a) When there is no memory for adding new name
b) When the name to be defined is already present in the table
c) Whenever a new name is added to the table
d) All of the mentioned

80) What is the output of this **C code?**

```
1. #include <stdio.h>

2. struct student
```

```
3.  {
4.      char a[];
5.  };
6.  int main()
7.  {
8.      struct student s;
9.      printf("%d", sizeof(struct student));
10.     return 0;
11. }
```

a) Compile time error
b) 8
c) 1
d) Varies

81) What is the output of this C code?

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      struct p
5.      {
6.          char *name;
7.          struct p *next;
8.      };
9.      struct p *ptrary[10];
10.     struct p p, q;
11.     p.name = "xyz";
12.     p.next = NULL;
```

```
13.     ptrary[0] = &p;
14.     q.name = (char*)malloc(sizeof(char)*3);
15.     strcpy(q.name, p.name);
16.     q.next = &q;
17.     ptrary[1] = &q;
18.     printf("%s\n", ptrary[1]->next->next->name);
19.     return 0;
20. }
```

a) Compile time error
b) Depends on the compiler.
c) Undefined behaviour
d) xyz

References:

1)http://www.sanfoundry.com/c-interview-questions-answers/