Chapter 11

Roll 2102021, Reg 10151

MULTIPLE CHOICE

11.1 (d) ~~Using dot notation, V.x~~ All of the above

11.2 (d) All of the above

11.3 (c) Unlike structures, union does not support nesting

11.4 (a) =

11.5 (d) compilation error

REVIEW QUESTIONS

11.1

(a) False
(b) True
(c) False
(d) True
(e) True
(f) True
(g) True
(h) True
(i) True
(j) False
(k) True
(l) False
(m) False
(n) False
(o) False

11.2

(a) tag name

(b) members

(c) typedef

(d) pointers

(e) union

(f) 0

(g) slack

(h) largest

11.3

(a) Incorrect. Tagname missing. It would be

struct abc : a, b, c;

(b) Incorrect. Semicolon missing.

(c) Incorrect. Struct keyword missing.

(d) Incorrect. Matrix size not given

(e) Correct

(f) Incorrect. B is outside braces.

(g) Correct.

(h) Incorrect. Brace was not used.

11.4

(e) a = b;

11.5 (d) struct item_bank items [10];

11.6 (a) correct

(b) correct

(c) Incorrect. the tagname is 'abc'.

(d) Correct

(e) Correct

11.7 Array is combination of data of same type where structured is user defined custom data type ~~or~~ and can contain multiple data types.

11.8 Template: Template is the blue print of a structure and tells how a structure is formed.

Struct keyword: struct keyword is used to define a structure and it is mandatory to form a structure.

Typedef keyword: Typedef is used to define a name for a variable. And can be used to make struct more covenient.

sizeof operator: Sizeof returns the size of a variable and can be used to count the elements number.

Tag name: Tag name is defined in structure so that it can be used later to make more similar data typed objects.

11.9

```
struct
{
  int number;
  float price;
}
```

Ans: Here the semicolon is missing and do not contain any tag name. So it may be inaccessible.

11.10

(a) Unions: Unions are used when only one element may be used and we want to save our storage.

(b) Bit fields: When we need to save memory.

(c) Sizeof: To determine the size.

11.11

(a) Array of structures

Ans: After defining a structure, we can make an array containing our custom data type like,

```
struct ABC a[10];
```

(b) Nested structures

Ans: When we use one structure inside an another then that is nested structure. for example

```
struct ABC {
    struct HELLO hi;
    .....
};
```

(c) Unions

Ans: Unions is a special data type that use same memory for multiple variable for efficiency.

11.12 To assign value we can,

(a) Use dot operator

```
ABC.a = 5;
```

(b) Assign while initializing,

```
struct ABC a = {5, 6, 7};
```

11.13 While initializing we can't directly assign any value. And the orders must match while initializing.

11.14 Slack byte is the unoccupied byte in structure. As it initializes to a default value and undefined. That's why we can't compare two structure directly.

11.15 (a) First method is to copy and send each members individually.

(b) Second method is to pass a copy of entire structure.

(c) Third method is to send the address and use pointer to use structure as argument.

11.16 Bit fields may varry from compiler to compilers and thus not recommended for all scenarios. And it can be used with static or extern.

11.17 Ans:

```
#include <stdio.h>
struct complex {
    float x, y;
};
int main()
{
    struct complex p = { 0.0, 1.1};
}
```

11.18 Ans: Compiler error!
A is not declared.

DEBUGGING EXCERCISE

11.1 (a) Correct

(b) Incorrect. It should be a1.b = 10.75;

(c) Incorrect. It should be int m = abc(a1.a + a1.b);

(d) Correct

(e) Correct

(f) Incorrect. Correct way is to use tag name first.

(g) Correct

(h) Incorrect. We can't compare structures.

11.2

Ans: Here we can't declare new array while using typedef. We have to omit studwise [10]; part.

INTERVIEW QUESTIONS

11.1 Ans: In terms of object size union saves more storage than structure. Because in union values are stored under same data type.

11.2 Ans: Postfix increment or decrement operator has the highest precedence in c programming while comma has the lowest.

11.3 Ans: Self referential structure means the condition when a member of a structure points to another member of the same structure.

11.4 Ans: To create new data types available options are struct, typedef, union and enumerate.

11.5 Ans: Nope, it is not a valid structure. A memeer of a structure cannot be an another same structure.