Structures, Unions, Enums

General Questions

1. How will you free the allocated memory?
   - A. remove(var-name);
   - B. free(var-name);
   - C. delete(var-name);
   - D. dalloc(var-name);

   Answer & Explanation

   Answer: Option B

2. What is the similarity between a structure, union and enumeration?
   - A. All of them let you define new values
   - B. All of them let you define new data types
   - C. All of them let you define new pointers
   - D. All of them let you define new structures

   Answer & Explanation

   Answer: Option B

Find Output of Program

1. What will be the output of the program?

   ```c
   #include<stdio.h>
   int main()
   {
   union a
   {
       int i;
       char ch[2];
   };
   union a u;
   u.ch[0]=3;
   u.ch[1]=2;
   printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i);
   return 0;
   }
   ```

   - A. 3, 2, 515
   - B. 515, 2, 3
   - C. 3, 2, 5
   - D. 515, 515, 4

   Answer & Explanation

   Answer: Option A

   Explanation:

   The system will allocate 2 bytes for the union.

   The statements `u.ch[0]=3; u.ch[1]=2;` store data in memory as given below.
2. What will be the output of the program?

#include<stdio.h>

int main()
{
    union var
    {
        int a, b;
    }
    union var v;
    v.a=10;
    v.b=20;
    printf("%d\n", v.a);
    return 0;
}

A. 10
B. 20
C. 30
D. 0

Answer & Explanation

Answer: Option B

3. What will be the output of the program?

#include<stdio.h>

int main()
{
    struct value
    {
        int bit1:1;
        int bit3:4;
        int bit4:4;
    } bit=(1, 2, 13);

    printf("%d, %d, %d\n", bit.bit1, bit.bit3, bit.bit4);
    return 0;
}

A. 1, 2, 13
B. 1, 4, 4
C. -1, 2, -3
D. -1, -2, -13

Answer & Explanation
Answer: Option C

Explanation:

Note the below statement inside the struct:

```c
int bit1:1;
```

-->

'int' indicates that it is a SIGNED integer.

For signed integers the leftmost bit will be taken for +/- sign.

**If you store 1 in 1-bit field:**

The left most bit is 1, so the system will treat the value as negative number.

The 2’s complement method is used by the system to handle the negative values.

Therefore, the data stored is 1. The 2's complement of 1 is also 1 (negative).

Therefore -1 is printed.

**If you store 2 in 4-bits field:**

Binary 2: 0010 (left most bit is 0, so system will treat it as positive value)

0010 is 2

Therefore 2 is printed.

**If you store 13 in 4-bits field:**

Binary 13: 1101 (left most bit is 1, so system will treat it as negative value)

Find 2's complement of 1101:

1's complement of 1101 : 0010
2's complement of 1101 : 0011 (Add 1 to the result of 1's complement)

0011 is 3 (but negative value)

Therefore -3 is printed.

4. What will be the output of the program in 16 bit platform (Turbo C under DOS) ?

```c
#include<stdio.h>

int main()
{
    struct value
    {
        int bit1:1;
        int bit3:4;
    }
```
int bit4:4;
}bit;
printf("%d\n", sizeof(bit));
return 0;
}

A. 1  
C. 4  

Answer & Explanation

Answer: Option B

Explanation:
Since C is a compiler dependent language, in Turbo C (DOS) the output will be 2, but in GCC (Linux) the output will be 4.

5. What will be the output of the program?

#include<stdio.h>
int main()
{
    enum days {MON=-1, TUE, WED=6, THU, FRI, SAT};
    printf("%d, %d, %d, %d, %d, %d\n", MON, TUE, WED, THU, FRI, SAT);
    return 0;
}

A. -1, 0, 1, 2, 3, 4  
B. -1, 2, 6, 3, 4, 5  
C. -1, 0, 6, 2, 3, 4  
D. -1, 0, 6, 7, 8, 9  

Answer & Explanation

Answer: Option D

6. What will be the output of the program?

#include<stdio.h>
int main()
{
    enum status {pass, fail, absent};
    enum status stud1, stud2, stud3;
    stud1 = pass;
    stud2 = absent;
    stud3 = fail;
    printf("%d %d %d\n", stud1, stud2, stud3);
    return 0;
}

A. 0, 1, 2  
B. 1, 2, 3  
C. 0, 2, 1  
D. 1, 3, 2  

Answer & Explanation

Answer: Option C

7. What will be the output of the program?

#include<stdio.h>
int main()
{
    int i=4, j=8;
    printf("%d, %d, %d\n", i|j&j|i, i|j&j|i, i^j);
    return 0;
}

A. 12, 12, 12  B. 112, 1, 12
C. 32, 1, 12  D. -64, 1, 12

Answer & Explanation

Answer: Option A

8. What will be the output of the program in Turbo C (under DOS)?

```c
#include<stdio.h>

int main()
{
    struct emp
    {
        char *n;
        int age;
    }
    struct emp e1 = {"Dravid", 23};
    struct emp e2 = e1;
    strupr(e2.n);
    printf("%s\n", e1.n);
    return 0;
}

A. Error: Invalid structure assignment  B. DRAVID
C. Dravid  D. No output

Answer & Explanation

Answer: Option B

9. What will be the output of the program in 16-bit platform (under DOS)?

```c
#include<stdio.h>

int main()
{
    struct node
    {
        int data;
        struct node *link;
    }
    struct node *p, *q;
    p = (struct node *) malloc(sizeof(struct node));
    q = (struct node *) malloc(sizeof(struct node));
    printf("%d, %d\n", sizeof(p), sizeof(q));
    return 0;
}

A. 2, 2  B. 8, 8
C. 5, 5  D. 4, 4

Answer: Option C

Saikat Banerjee
10. What will be the output of the program?

```c
#include<stdio.h>

int main()
{
    struct byte
    {
        int one:1;
    }
    struct byte var = {1};
    printf("%d\n", var.one);
    return 0;
}
```

A. 1  
B. -1  
C. 0  
D. Error

**Answer: Option B**

11. What will be the output of the program?

```c
#include<stdio.h>

int main()
{
    enum days {MON=-1, TUE, WED=6, THU, FRI, SAT};
    printf("%d, %d, %d, %d, %d, %d\n", ++MON, TUE, WED, THU, FRI, SAT);
    return 0;
}
```

A. -1, 0, 1, 2, 3, 4  
B. Error  
C. 0, 1, 6, 3, 4, 5  
D. 0, 0, 6, 7, 8, 9

**Answer: Option B**

12. What will be the output of the program?

```c
#include<stdio.h>

struct course
{
    int courseno;
    char coursename[25];
};

int main()
{
    struct course c[] = { {102, "Java"},
                         {103, "PHP"},
                         {104, "DotNet"}     };
    printf("%d ", c[1].courseno);
```
13. What will be the output of the program given below in 16-bit platform?

```c
#include<stdio.h>

int main()
{
    enum value{VAL1=0, VAL2, VAL3, VAL4, VAL5} var;
    printf("%d\n", sizeof(var));
    return 0;
}
```

A. 1  
B. 2  
C. 4  
D. 10

**Answer & Explanation**  
**Answer:** Option B

**Point Out Errors**

1. Point out the error in the program?

```c
struct emp
{
    int ecode;
    struct emp *e;
};
```

A. Error: in structure declaration  
B. Linker Error  
C. No Error  
D. None of above

**Answer & Explanation**  
**Answer:** Option C

**Explanation:**

This type of declaration is called as self-referential structure. Here *e is pointer to a struct emp.

2. Point out the error in the program?

```c
typedef struct data mystruct;
struct data
{
    int x;
    mystruct *b;
};
```
A. Error: in structure declaration
B. Linker Error
C. No Error
D. None of above

Answer & Explanation

Answer: Option C

Explanation:
Here the type name `mystruct` is known at the point of declaring the structure, as it is already defined.

3. Point out the error in the program?

```c
#include<stdio.h>

int main()
{
    struct a
    {
        float category:5;
        char scheme:4;
    
    }
    printf("size=%d", sizeof(struct a));
    return 0;
}
```

A. Error: invalid structure member in printf
B. Error in this float category:5; statement
C. No error
D. None of above

Answer & Explanation

Answer: Option B

Explanation:
Bit field type must be `signed int` or `unsigned int`.
The char type: `char scheme:4;` is also a valid statement.

4. Point out the error in the program?

```c
struct emp
{
    int ecode;
    struct emp e;
};
```

A. Error: in structure declaration
B. Linker Error
C. No Error
D. None of above

Answer & Explanation
Answer: Option A

Explanation:

The structure `emp` contains a member `e` of the same type. (i.e) `struct emp`. At this stage compiler does not know the size of structure.

5. Point out the error in the program?

```c
#include<stdio.h>

int main()
{
    struct emp
    {
        char name[20];
        float sal;
    }
    struct emp e[10];
    int i;
    for(i=0; i<=9; i++)
        scanf("%s %f", e[i].name, &e[i].sal);
    return 0;
}
```

A. Error: invalid structure member  
B. Error: Floating point formats not linked  
C. No error  
D. None of above

**Answer & Explanation**

**Answer: Option B**

**Explanation:**

At run time it will show an error then program will be terminated.

Sample output: Turbo C (Windows)

c:\>myprogram

Sample  
12.123

scanf : floating point formats not linked  
Abnormal program termination

6. Point out the error in the program?

```c
#include<stdio.h>
#include<string.h>

void modify(struct emp*);

struct emp
{
    char name[20];
    int age;
};
```
int main()
{
    struct emp e = {"Sanjay", 35};
    modify(&e);
    printf("%s %d", e.name, e.age);
    return 0;
}

void modify(struct emp *p)
{
    p->age = p->age + 2;
}

A. Error: in structure
B. Error: in prototype declaration unknown struct emp
C. No error
D. None of above

Answer & Explanation

Answer: Option B

Explanation:
The `struct emp` is mentioned in the prototype of the function `modify()` before declaring the structure. To solve this problem declare `struct emp` before the `modify()` prototype.

7. Point out the error in the program in 16-bit platform?

#include<stdio.h>

int main()
{
    struct bits
    {
        int i:40;
    }bit;
    printf("%d\n", sizeof(bit));
    return 0;
}

A. 4
B. 2
C. Error: Bit field too large
D. Error: Invalid member access in structure

Answer & Explanation

Answer: Option C

8. Point out the error in the program?

#include<stdio.h>

int main()
{
    union a
    {
        int i;
        char ch[2];
    };

A. Error: invalid union declaration
B. Error: in Initializing z2
C. No error
D. None of above

Answer: Option B

9. Point out the error in the program?

```c
#include<stdio.h>

int main()
{
    struct emp
    {
        char n[20];
        int age;
    }
    struct emp e1 = {"Dravid", 23};
    struct emp e2 = e1;
    if(e1 == e2)
    {
        printf("The structure are equal");
        return 0;
    }
}
```

A. Prints: The structure are equal
B. Error: Structure cannot be compared using '=='
C. No output
D. None of above

Answer: Option B

10. Point out the error in the program?

```c
#include<stdio.h>

int main()
{
    struct bits
    {
        float f:2;
    }bit;

    printf("%d\n", sizeof(bit));
    return 0;
}
```

A. 4
B. 2
C. Error: cannot set bit field for float
D. Error: Invalid member access in structure

Answer & Explanation
Answer: Option C

11. Point out the error in the program?

```c
#include<stdio.h>

int main()
{
    struct emp
    {
        char name[25];
        int age;
        float bs;
    };
    struct emp e;
    e.name = "Suresh";
    e.age = 25;
    printf("%s %d\n", e.name, e.age);
    return 0;
}
```

A. Error: Lvalue required/incompatible types in assignment
B. Error: invalid constant expression
C. Error: Rvalue required
D. No error, Output: Suresh 25

Answer & Explanation

Answer: Option A

Explanation:

We cannot assign a string to a struct variable like `e.name = "Suresh"` in C.

We have to use `strcpy(char *dest, const char *source)` function to assign a string.

Ex: `strcpy(e.name, "Suresh");`

---

**Point Out Correct Statements**

1. Which of the following statements correct about the below program?

```c
#include<stdio.h>

int main()
{
    struct emp
    {
        char name[25];
        int age;
        float sal;
    };
    struct emp e[2];
    int i=0;
    for(i=0; i<2; i++)
        scanf("%s %d %f", e[i].name, &e[i].age, &e[i].sal);
    for(i=0; i<2; i++)
        scanf("%s %d %f", e[i].name, e[i].age, e[i].sal);
```
A. Error: `scanf()` function cannot be used for structures elements.
B. The code runs successfully.
C. Error: Floating point formats not linked Abnormal program termination.
D. Error: structure variable must be initialized.

**Answer & Explanation**

**Answer:** Option C

2. Which of the following statements correct about the below program?

```c
#include<stdio.h>
int main()
{
    union a
    {
        int i;
        char ch[2];
    }
    union a u1 = {512};
    union a u2 = {0, 2};
    return 0;
}
```

1: `u2` CANNOT be initialized as shown.
2: `u1` can be initialized as shown.
3: To initialize `char ch[]` of `u2` , `.` operator should be used.
4: The code causes an error 'Declaration syntax error'

**A.** 1, 2
**B.** 2, 3
**C.** 1, 2, 3
**D.** 1, 3, 4

**Answer & Explanation**

**Answer:** Option C

3. Which of the following statements correctly assigns 12 to `month` using pointer variable `pdt`?

```c
#include<stdio.h>
struct date
{
    int day;
    int month;
    int year;
};
int main()
{
    struct date d;
    struct date *pdt;
    pdt = &d;
    return 0;
}
```

**A.** `pdt.month = 12`
**B.** `&pdt.month = 12`
**C.** `d.month = 12`
**D.** `pdt->month = 12`

**Answer & Explanation**

**Answer:** Option D
4. Which of the following statements correct about the below code?
maruti.engine.bolts=25;

A. Structure bolts is nested within structure engine.
B. Structure engine is nested within structure maruti.
C. Structure maruti is nested within structure engine.
D. Structure maruti is nested within structure bolts.

Answer: Option B

True / False Questions

1. A union cannot be nested in a structure
A. True
B. False

Answer: Option B

2. Nested unions are allowed
A. True
B. False

Answer: Option A

3. Bit fields CANNOT be used in union.
A. True
B. False

Answer: Option B

Explanation:
The following is the example program to explain "using bit fields inside an union".

```c
#include<stdio.h>

union Point
{
    unsigned int x:4;
    unsigned int y:4;
    int res;
};

int main()
{
    union Point pt;
}
```
4. One of elements of a structure can be a pointer to the same structure.
   A. True  B. False
   Answer & Explanation
   Answer: Option A

5. A structure can be nested inside another structure.
   A. True  B. False
   Answer & Explanation
   Answer: Option A

Explanation:

6. Which of the following statement is True?
   A. User has to explicitly define the numeric value of enumerations
   B. User has a control over the size of enumeration variables.
   C. Enumeration can have an effect local to the block, if desired
   D. Enumerations have a global effect throughout the file.
   Answer & Explanation
   Answer: Option C

7. The '.' operator can be used access structure elements using a structure variable.
   A. True  B. False
   Answer & Explanation
   Answer: Option A

8. Union elements can be of different sizes.
   A. True  B. False
   Answer & Explanation
   Answer: Option A

9. A structure can contain similar or dissimilar elements
   A. True  B. False
   Saikat Banerjee
10. The ‘->’ operator can be used to access structures elements using a pointer to a structure variable only
   A. True
   B. False
   
   Answer: Option A

11. It is not possible to create an array of pointer to structures.
   A. True
   B. False
   
   Answer: Option B

**Yes / No Questions**

1. If the following structure is written to a file using `fwrite()`, can `fread()` read it back successfully?

   ```c
   struct emp
   {
       char *n;
       int age;
   };
   struct emp e="Sbdsisaikat", 15);
   FILE *fp;
   fwrite(&e, sizeof(e), 1, fp);
   ```
   A. Yes
   B. No
   
   Answer: Option B

   **Explanation:**
   Since the structure contain a `char` pointer while writing the structure to the disk using `fwrite()` only the value stored in pointer `n` will get written. so `fread()` fails to read.

2. size of union is size of the longest element in the union
   A. Yes
   B. No
   
   Answer: Option A

3. The elements of union are always accessed using & operator
   A. Yes
   B. No
   
   Answer: Option B
4. Will the following code work?

```c
#include<stdio.h>
#include<alloc.h>

struct emp
{
    int len;
    char name[1];
};

int main()
{
    char newname[] = "Rahul";
    struct emp *p = (struct emp *) malloc(sizeof(struct emp) - 1 + strlen(newname)+1);

    p->len = strlen(newname);
    strcpy(p->name, newname);
    printf("%d %s\n", p->len, p->name);
    return 0;
}
```

A. Yes  B. No

Answer: Option A

Explanation:
The program allocates space for the structure with the size adjusted so that the `name` field can hold the requested name.

5. A pointer union CANNOT be created

A. Yes  B. No

Answer: Option B

6. Is there easy way to print enumeration values symbolically?

A. Yes  B. No

Answer: Option B

Explanation:
You can write a function of your own to map an enumeration constant to a string.

7. By default structure variable will be of auto storage class
8. Is it necessary that the size of all elements in a union should be same?
A. Yes
B. No
Answer & Explanation

Answer: Option B

9. Can we have an array of bit fields?
A. Yes
B. No
Answer & Explanation

Answer: Option B

10. Will the following declaration work?

```c
typedef struct s
{
    int a;
    float b;
}s;
```
A. Yes
B. No
Answer & Explanation

Answer: Option A

11. Can a structure can point to itself?
A. Yes
B. No
Answer & Explanation

Answer: Option A

Explanation:

A structure pointing to itself is called self-referential structures.

12. If a char is 1 byte wide, an integer is 2 bytes wide and a long integer is 4 bytes wide then will the following structure always occupy 7 bytes?

```c
struct ex
{
    char ch;
    int i;
    long int a;
};
```
A. Yes
B. No

Answer & Explanation

**Answer:** Option B

**Explanation:**

A compiler may leave holes in structures by padding the first char in the structure with another byte just to ensure that the integer that follows is stored at an location. Also, there might be 2 extra bytes after the integer to ensure that the long integer is stored at an address, which is multiple of 4. Such alignment is done by machines to improve the efficiency of accessing values.