1.
```
#include<stdio.h>
struct s1{
    char a[4];
    char *p;
}o = {"DAC","DMC"};
int main(void)
{
    o.p = o.a+1;
    printf("%c %s\n",*o.p,o.a);
    return 0;
}
```

A. M DAC
B. A DAC
C. A DMC
D. M DMC

Answer: B

2.Consider 32 bit compilation.
```
#include<stdio.h>
struct s2 {
    char *cp;
    struct s1{
        char a[4];
        char *p;
    }o1;
}o2;
int main(void)
{
    printf("%d %d %d\n",sizeof(struct s2) , sizeof(o2) ,
                        sizeof(o2.o1));
    return 0;
}
```
A. 24 24 16
B. 12 12 8
C. Both A and B
D. None of the above

Answer: B

3.

```
#include<stdio.h>
struct s2 {
    char *cp;
    struct s1{
        char a[4];
        char *p;
    }o1;
}o2 = {"DAC", "DESD", "DMC"};

int main(void)
{
    printf("%s %s\n",++o2.cp,++o2.o1.p);
    return 0;
}
```

A. AC MC
B. AC ESD
C. MC AC
D. ESD MC

Answer: A

4.

```
#include<stdio.h>
int main(void)
{
    struct s{
        char *p;int i;
        struct s *sp;
    }a[] = {"abcd",1,a+1,"efgh",2,a+2,"ijkl",3,a}, *p;
    p = a;
    printf("%s %s %s\n",a[0].p,p->p,a[2].sp->p);
    return 0;
}
```

A. abcd abcd ijkl
B. abcd efgh ijkl
C. abcd abcd efgh
D. abcd abcd abcd

Answer: D

5.

```
#include<stdio.h>
struct s {
    int i;
    struct s obj;
}s1;
int main(void)
{
    s1.i = 100;
    s1.obj = s1;
    printf("%d",s1.obj.i);
    return 0;
}
```

A. 100
B. Garbage value
C. Compiler error
D. Run time error

Answer: C

6.

```
#include<stdio.h>
union
{
    short i;
    char c;
}u;
int main(void)
{
    u.c = 'D';
    u.i = 0x0041;
    printf("%d %c",sizeof(u),u.c);
    return 0;
}
```

A. 2 A
B. 2 D
C. 3 A
D. 3 D

Answer: A

7.

```
#include<stdio.h>
union u{
    int i;
    char c[4];
};
int main(void)
{
    union u u1;
    u1.i=0;
    u1.c[1] = u1.c[2] = 'F';
    printf("%s",u1.c);
    return 0;
}
```

A. no outout return value from main function is zero
B. Garbage character followed by 'FF'
C. FF
D. Compile Error

Answer: A

8. consider 32 bit compilation.

```
#include<stdio.h>
#pragma pack(1)
        struct
        {
            char ca[10];
            union u
            {
                int i;
                char c;
                long int l;
             }u1;
        }s1;
int main(void)
{
    printf("%d", sizeof(s1) + sizeof(s1.u1));
    return 0;
}
```

A. 20
B. 18
C. 23
D. 26

Answer: B

9.
```c
#include<stdio.h>
int main(void)
{
    enum colours {RED,BLACK,WHITE=5,YELLOW,BLUE,GREY};
    printf("%d %d %d %d",RED,YELLOW,BLUE,GREY);
    return 0;
}
```

A. 1 2 3 4
B. 0 2 3 4
C. 1 6 7 8
D. 0 6 7 8

Answer: D

10.
```c
#include<stdio.h>
int main(void)
{
    enum choice {CH1, CH2, CH3};
    enum choice ch1, ch2, ch3;
    ch1 = CH1;
    ch2 = CH3;
    ch3 = CH2;
    printf("%d, %d, %d,", ch1, ch2, ch3);
    return 0;
}
```

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

Answer: C

11.

```
#include<stdio.h>
int main(void)
{
    enum choice {CH1, CH2, CH3};
    enum choice ch1, ch2, ch3;
    ch1 = CH1;
    ch2 = CH3;
    ch3 = ch2-ch1;

    printf("%d %d", sizeof(enum choice), ch3);
    return 0;
}
```

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

Answer: A

12.

```
#include<stdio.h>
int main(void)
{
    typedef int int_t;
    int_t *iptr;
    int ival = 60;
    iptr = &ival;
    printf("%d",*iptr);
    return 0;
}
```

A. 60
B. Compiler error
C. Linker error
D. Runtime error

Answer: A

13.

Select correct answer
typedef int (*funptr)(int, int);

A. funptr is a function pointer
B. funptr can be used as type
C. Both A and B
D. None of the above

Answer: C

14.

```
#include<stdio.h>
int main(void)
{
    typedef struct {
        int val;
        test_t *ptr;
    }test_t;
    test_t obj = { 25, &obj};
    printf("%d",obj.ptr->val);
    return 0;
}
```

A. 25
B. Compiler time error
C. Run time error
D. None of the above

Answer: B

15.

```
#include<stdio.h>
struct demo
{
    int d1:2;
    int d2:3;
};
```

```c
int main(void)
{
    struct demo ds;
    ds.d1 = 1;ds.d2 = 3;
    printf("%d %d ", ds.d1,ds.d2);
    ds.d1 = 2;ds.d2 = 7;
    printf("%d %d", ds.d1,ds.d2);
    return 0;
}
```

A. 1 3 2 7
B. 1 3 -2 -1
C. Compiler error
D. None of the above

Answer: B

16.

```c
#include<stdio.h>
struct time {
    int ss:7;
    int mm:7;
    int hh:4;
};

int main(void)
{
    struct time t1;
    printf("%d %d", sizeof(t1), sizeof(t1.ss));
    return 0;
}
```

A. 4
B. Compiler error
C. Runtime error
D. None of the above

Answer: B