

1.

```
#include <stdio.h>
typedef struct Cell
{
    int isParent;
    Cell child;
}Cell;
int main(void)
{
    Cell parent,child;
    parent.child=child;
    return 0;
}
```

- A. successfully complied
- B. successfully complied and executed
- C. Compile time error
- D. none of above

Answer: C

2.

```
#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
- A. 12 12 8
- C. Both A and B
- D. None of the above

Answer: B

3.

```
#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

4.

```
#include <stdio.h>
struct student
{
    int rollno;
    char name[10];
    char *subject;
    char division;
    short int std;
    struct student *point;
};

int main(void)
{
    struct student s;
    struct student m;

    s.subject = m.subject = "C++";
    m.point = &s;
    (m.point)->subject = "CPP";
    printf("%s\t%s\t", s.subject, m.subject);

    return 0;
}
```

- A. CPP C++
- B. CPP CPP
- C. C++ C++
- D. Runtime error

Answer: A

5.

```
#include <stdio.h>
#include <string.h>
struct Test
{
    char str[9];
};
```

```
int main(void)
{
    struct Test st1, st2;

    strcpy(st1.str, "CSharp");
    st2 = st1; st1.str[0] = 'J';
    printf("%s \t %s",st2.str,st1.str);

    return 0;
}
```

- A. CSharp JSharp
- B. JSharp CSharp
- C. Segmentation Fault
- D. Compile Time Error
- E. CSharp CSharp

Answer: A