

Day10_Poll_Questions

1.

```
#include <stdio.h>
int rec(int);
int main(void)
{
    static int val = 8;
    printf("%d ", rec(val));

    return 0;
}
int rec(int val)
{
    static int sal = 10;
    if(val)
        return sal + rec(--val);
}
```

- A. 80
- B. Compile time error
- C. Runtime time error(exit status -1)
- D. 72
- E. 88

Answer: C

2.

Identify the incorrect statement

- I. Automatic variables are automatically initialized to 0
- II. Static variables are automatically initialized to 0
- III. The address of a register variable is not accessible
- IV. Static variables cannot be initialized with local variables.

- A. I
- B. I,II
- C. II,III
- D. IV

Answer :A

3.

```
#include <stdio.h>
int ext = 300;
int main(void)
{
    extern int ext;
    printf("Ext = %d ", ext);
    extfun();
    return 0;
}
int ext = 100;
int extfun(void)
{
    int ext = 200;
    printf("%d\n",ext);
    return ext;
}
```

- A. Ext = 100 200
- B. Ext = 300 200
- C. Compile time error
- D. Run time error

Answer: C

4.

What is the meaning of using **extern** before declaration ?
For example following function sum is made **extern**

```
extern int subtract(int x, int y, int z)
{
    return (x + y + z);
}
```

- A. Function is made globally available
- B. extern means nothing ,subtract() is same without **extern** keyword in function.
- C. Function need not be declared before its use
- D. Function is made local to the file.

Answer :B

5.

```
#include <stdio.h>
void f();
int main()
{
    int a=10;
    a = f();
    printf("%d", a);

    return 0;
}
void f()
{
    printf("Hi");
    return ;
}
```

- A. Error:function f does not **return** any value so we can not catch in variable a.
- B. Hi
- C. **10**
- D. Hi **10**

Answer :A