

DEBUGGING AND REVERSE CODING

1)

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

```
int main(void) {
```

```
    int a[5] = {1,2,3,4,5};
    int *ptr = (int*)&a+1;
    printf("%d %d", *(a+2), *(ptr-2));
    return 0;
```

```
}
```

a)3,4 b)3,3 c)Garbage Value d)Segmentation Fault

2)

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

```
char *a[] = {"Abacus", "Here", "We", "Go"};
char **b[] = {a+3, a+2, a+1, a};
char ***c = b;
```

```
int main()
```

```
{
```

```
    printf("%s ", **++c);
    printf("%s ", c[-1][-1]);
    return 0;
```

```
}
```

a)We We b)We Abacus c)We Here d)None of these

3)

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

```
#define HERE(n, a, i, m) m###a###i###n
```

```
#define ABACUS HERE(n, a, i, m)
```

```
int ABACUS()
```

```
{
```

```
    printf("Welcome");
    return 0;
```

```
}
```

a)Welcome b)Compile Time Error c)Run Time Error

4)

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

```
void call(int s, ...)
```

```
{
```

```
    printf("%d ", s);
```

```
}
```

```
int main()
```

```
{
```

```
    call(3, 6, 7, 8);
```

```
    call(5, 6, 9);
```

```
    return 0;
```

```
}
```

A)3 6 b)3 5 c)Compile Error d)Run time Error

5)

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

```
struct abc
```

```
{
```

```
    int x, y, z;
```

```
};
```

```
int main()
```

```
{
```

```
    struct abc d = {.y = 0, .z = 1, .x = 2};
```

```
    printf("%d %d %d", d.x, d.y, d.z);
```

```
    return 0;
```

```
}
```

A)0 1 2 b)1 0 2 c)2 0 1 d)Compile Error

6)

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

```
int fun(char *b)
```

```
{
```

```
    char *a = b;
```

```
    while(*++a);
```

```
    return (a-b);  
}
```

```
int main()  
{  
    char *str = "Dracarys";  
    printf("%d", fun(str));  
    return 0;  
}
```

a)10 b)9 c)8 d)Compile Error

7)

```
#include "stdio.h"
```

```
int * a;
```

```
int main()  
{  
    int * b = NULL;
```

```
    if(a == b)  
    {  
        printf("Not Equal");  
    }  
    else  
    {  
        printf("Equal");  
    }
```

```
    return 0;  
}
```

a)Not Equal b)Equal c)Compile Time Error

8)

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

```
#define SIZE(arr) sizeof(arr) / sizeof(*arr);
```

```
void fun(int* arr, int n)
```

```
{  
    int i;  
    *arr += *(arr + n - 1) += 10;  
}
```

```
void printArr(int* arr, int n)
```

```

{
    int i;
    for(i = 0; i < n; ++i)
        printf("%d ", arr[i]);
}

```

```

int main()
{
    int arr[] = {1, 2, 7};
    int size = SIZE(arr);
    fun(arr, size);
    printArr(arr, size);
    return 0;
}

```

A)7 2 17 b)1 2 17 c)18 2 17 d)Compile Time Error

9)

```

#include<stdio.h>
int main()
{
    char a = 125;
    a = a+149;
    printf("%d", a);
    return 0;
}

```

a)Out of Range Exc b)149 c)18 d)-127

10)

```

#include<stdio.h>
int main()
{
    float x = 0.1;
    if ( x == 0.1 )
        printf("GOT");
    else if (x == 0.1f)
        printf("StrangerThings");
    else
        printf("West World");
}

```

a)GOT b)StrangerThings c)West World d)Error