

PART-B

2. a) A null pointer has a value which indicates the compiler that pointer does not contain any valid object. It is used to -
- 1) initialize pointers
 - 2) indicate errors in returning pointers from a function.
 - 3) represent condition such as end of list of unknown length
- 2 b) Dangling pointers occurs at time of object destruction i.e when the object located by the pointer is delocated or destroyed.

```
Ex # include <stdlib.h>
int main()
{
    int *ptr = (int*) malloc(sizeof(int))
    ptr = 560;
    // ptr does not point at 560.
```

2 c)

```
# include <stdio.h>
int main() {
    int a, b; *ptr1, *ptr2, **dp1, **dp2
    a = 6; b = 9;
    *ptr1 = a; **dp1 = *ptr1
    *ptr2 = b; **dp2 = *ptr2

    int sum = *dp1 + **dp2
    printf("%d", sum)
```

Output
=> 15

4a)

```
int f(int p, int n) {  
    if (n <= 1) return 0;  
    else return max(f(p+1, n-1), p[p] - p[1]);  
}
```

```
int main()
```

```
{  
    int a[] = {3, 5, 2, 6, 4};  
    printf("%d", f(a, 5));  
}
```

input \Rightarrow for function f is array a and 5

output $\Rightarrow 3$

4(b)

```
void fb(int n) {  
    if (n == 1) printf("0");  
    if (n == 2) printf("1");  
    if (n > 2)  
        return fb(n-2);  
}
```

```
void fb(int n) {  
    if (n == 1) return 0;  
    if (n == 2) return 1;  
    if (n > 2)  
        return fb(n-2);  
}
```

```
int main() {  
    int n;  
    scanf("%d", &n);  
    int f = fb(n);  
    printf("%d", f);  
}
```