LAB-11

Exercise:

1) Implement the above code and paste the screen shot of the output.

PROGRAM:

```c
#include <stdio.h>
#include <stdlib.h>  // for exit()

int main() {
    int base[20], limit[20], n, i, pa, logical_address, segment;

    printf("\nProgram for Segmentation");
    printf("\nEnter the number of segments: ");
    scanf("%d", &n);

    printf("\nEnter the base address and limit for each segment:\n");
    for (i = 0; i < n; i++) {
        printf("Segment %d Base: ", i);
        scanf("%d", &base[i]);
        printf("Segment %d Limit: ", i);
        scanf("%d", &limit[i]);
    }

    printf("\nEnter the segment number: ");
    scanf("%d", &segment);
    printf("Enter the offset (logical address): ");
    scanf("%d", &logical_address);

    if (segment >= n || segment < 0) {
        printf("\nInvalid segment number.\n");
        return 0;
    }

    if (logical_address < limit[segment]) {
        pa = base[segment] + logical_address;
        printf("\n\tSegment\t Base\t Logical\t Physical\n");
        printf("\t%d\t %d\t %d\t\t %d\n", segment, base[segment], logical_address,
pa);
    } else {
        printf("\nOffset exceeds the limit of the segment.\n");
    }

    return 0;
}
```

OUTPUT:

```
PS D:\OS labs> cd "d:\OS labs\" ; if ($?) { gcc lab_11.c -o lab_11 } ; if ($?) { .\lab_11 }

  Program for Segmentation
  Enter the number of segments: 3

  Enter the base address and limit for each segment:
  Segment 0 Base: 100
  Segment 0 Limit: 50
  Segment 1 Base: 200
  Segment 1 Limit: 30
  Segment 2 Base: 300
  Segment 2 Limit: 40

  Enter the segment number: 1
  Enter the offset (logical address): 20

          Segment  Base    Logical         Physical
          1        200     20              220
PS D:\OS labs>
```