**CT-353 Talal Khan DT-22043**

**LAB-11**

**Exercise:**

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

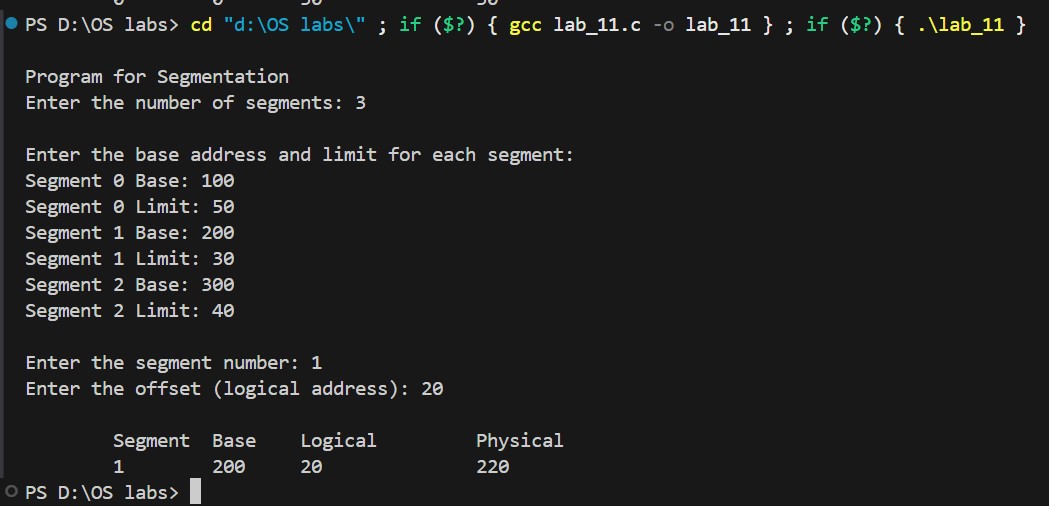
**PROGRAM:**

|  |
| --- |
| #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;  } |

**LAB-11 (MEMORY MANAGEMENT TECHNIQUES)**

**CT-353 Ezaan Khan DT-22046**

**OUTPUT:**



**LAB-11 (MEMORY MANAGEMENT TECHNIQUES)**