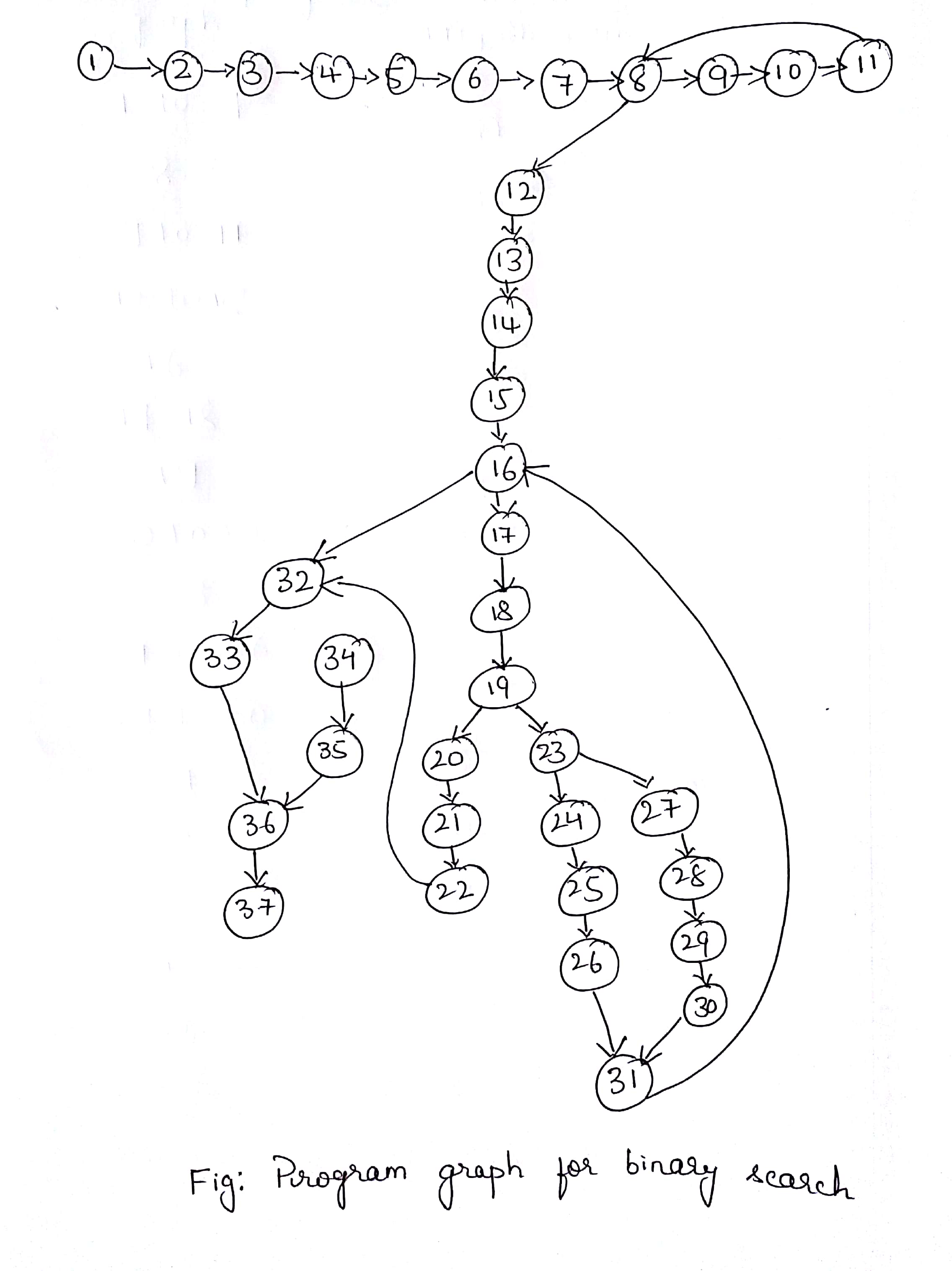
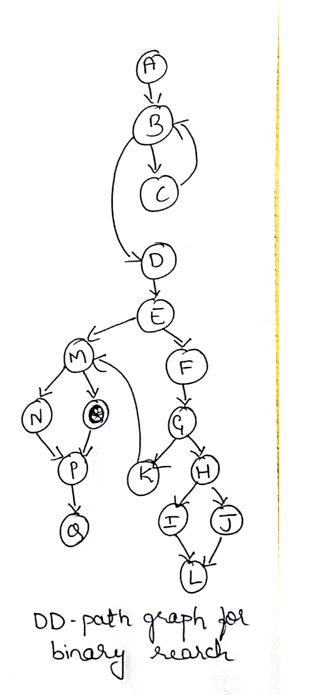
**10. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.**

1. #include<stdio.h>
2. int main()
3. {
4. int a[20],n,low,high,mid,key,i,flag=0;
5. printf("Enter the value of n:\n");
6. scanf("%d",&n);
7. printf("Enter %d elements in ASCENDING order\n",n);
8. for(i=0;i<n;i++)
9. {
10. scanf("%d",&a[i]);
11. }
12. printf("Enter the key element to be searched\n");
13. scanf("%d",&key);
14. low=0;
15. high=n-1;
16. while(low<=high)
17. {
18. mid=(low+high)/2;
19. if(a[mid]==key)
20. {
21. flag=1;
22. break; }
23. else if(a[mid]<key)
24. {
25. low=mid+1;
26. }
27. else
28. {
29. high=mid-1;
30. }
31. }
32. if(flag==1)
33. printf("Successful search\n Element found at Location %d\n",mid+1);
34. else
35. printf("Key Element not found\n");
36. return 0;
37. }



|  |  |
| --- | --- |
| **NODES IN PROGRAM GRAPH** | **NODES IN DD PATH GRAPH** |
| 1-7 | A |
| 8 | B |
| 9-11 | C |
| 12-15 | D |
| 16 | E |
| 17-18 | F |
| 19 | G |
| 20-22 | K |
| 23 | H |
| 24-26 | I |
| 27-30 | J |
| 31 | L |
| 32 | M |
| 33 | N |
| 34-35 | O |
| 36 | P |
| 37 | Q |



BINARY SEARCH: BASIS PATH TESTING

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TC ID** | **FLIP AT NODE** | **PATHS** | **INPUT** | | **EXPECTED OUTPUT** | **REMARKS** |
| **Array a[ ]** | **KEY** |
| **1** | **Baseline Path** | **P1:** ABCBDEFGHILEFGKMNPQ | {10,20,30} | 30 | Successful search Element found at location, 3 | Valid path |
| **2** | **H** | **P2:** ABCBDEFGHJLEFGKMNPQ | {10,20,30} | 10 | Successful search Element found at location, 1 | Valid path |
| **3** | **G** | **P3:** ABCBDEGKMNPQ | {10,20,30} | 20 | Successful search Element found at location, 2 | Valid path |
| **4** | **B** | **P4:**ABDEFGHILEFGKMN PQ | Empty | Any key | - | **Infeasible path.**  B to D means no elements entered and E to F indicates low<=high is true, which is not possible. |
| **5** | **E** | **P5:** ABCBDEMNPQ | {10,20,30} | Any key | - | **Infeasible path.**  B to C means elements are entered and E to M indicates low<=high is false, which is not possible. |
| **6** | **M** | **P6:**ABCBDEFGHILEFGKMOPQ | {10,20,30} | Any key | - | **Infeasible path.**  G to K means flag=1 and M to O indicates flag=0, which is not possible. |