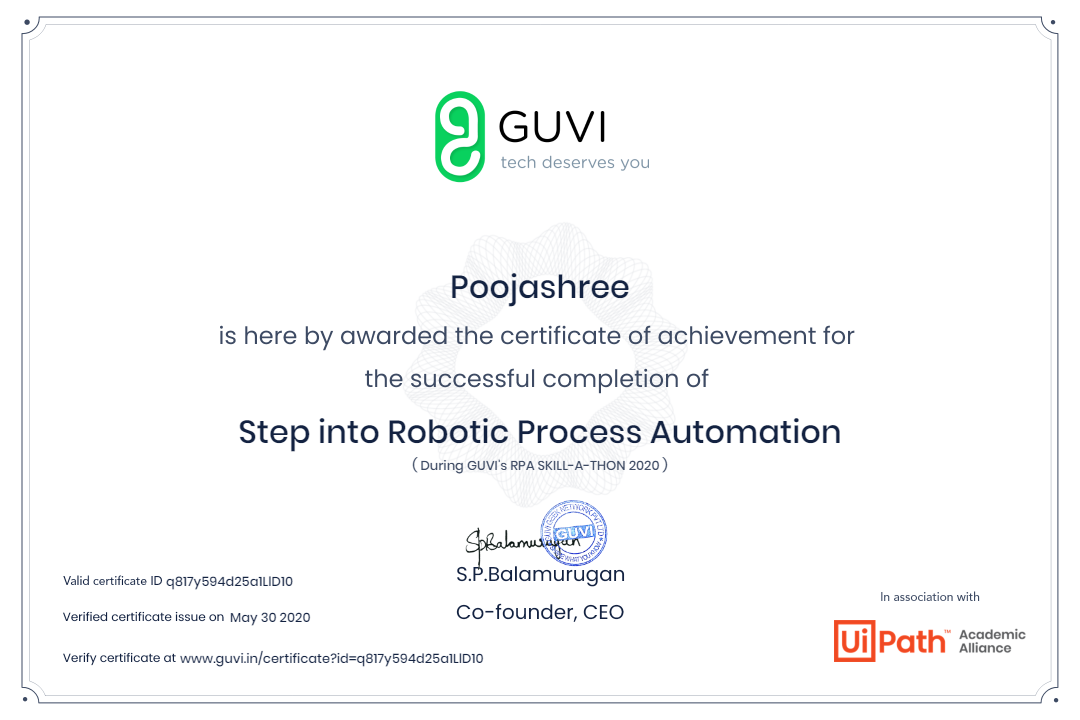
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **30-5-2020** | | | | | **Name:** | **poojashree** | |
| **Sem & Sec** | **8th sem A sec** | | | | | **USN:** | **4al16cs065** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | |  | | | | | | |
| **Max. Marks** | |  | | **Score** | | |  | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Step into robotic process automation** | | | | | | | |
| **Certificate Provider** | | | **GUVI** | | **Duration** | | | **3.5hr** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  **1**.  **haunted problem** | | | | | | | | |
| **Status:completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Poojashree** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

**Online test**

**-----------------not there-----------------**

**Certification course**



**Coding**

**Program 1**

1. *#include*<stdio.h>
2. *#include*<stdlib.h>
4. int age[100010];
5. int c[100010];
6. int a[100010];
7. int b[100010];
9. int compare(int \*a, int\*b)
10. {
11. return \*a-\*b;
12. }
14. int binarysearch(int n,int low,int high)
15. { int mid;
16. if (low > high)
17. return -1;
18. mid = (low + high)/2;
19. if(n == age[mid])
20. {
21. return mid;
22. }
23. if(n < age[mid])
24. { high = mid - 1;
25. return binarysearch(n,low,high);
26. }
27. if(n > age[mid])
28. { low = mid + 1;
29. return binarysearch(n,low,high);
30. }
31. }

34. int main()
35. {
36. int n, m, i, j, max[2];


40. scanf("%d%d", &n, &m);
42. for(i=0; i<n; i++)
43. {
44. scanf("%d", &a[i]);
45. b[i]=a[i];
46. }
48. qsort(b, n, sizeof(int), compare);
50. age[0]=b[0];
51. j=1;
53. for(i=1; i<n; i++)
54. {
55. if(b[i]!=age[j-1])
56. {
57. age[j]=b[i];
58. j++;
59. }
60. }
61. int M=j;
63. for(i=0; i<M; i++)
64. {
66. c[i]=0;
67. }
68. printf("\n");
70. max[0]=-1;
71. max[1]=-1;
73. for(i=0; i<n; i++)
74. {

77. int x = binarysearch(a[i], 0, M);
78. c[x]++;
80. if(c[x]>max[0])
81. {
82. max[0]=c[x];
83. max[1]=age[x];
84. }
85. else if(c[x]==max[0])
86. {
87. if(max[1]<age[x])
88. max[1]=age[x];
89. }
91. printf("%d %d\n", max[1], max[0]);
92. }
93. return 0;

}