

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  struct stack
5  {
6      int top;
7      int capacity;
8      int *array;
9  };
10
11 struct heightindexstack
12 {
13     struct stack *height;
14     struct stack *index;
15 };
16
17 struct stack* createStack(int capacity)
18 {
19     struct stack *s=(struct stack *)malloc(sizeof(struct stack));
20     s->top=-1;
21     s->capacity=capacity;
22     s->array=(int *)malloc(capacity*sizeof(int));
23     return s;
24 }
25
26 struct heightindexstack * createHeightIndexStack()
27 {
28     struct heightindexstack *s=(struct heightindexstack *)malloc(sizeof(struct
heightindexstack));
29     s->height=createStack(10);
30     s->index=createStack(10);
31     return s;
32 };
33
34 int isEmpty(struct stack *s)
35 {
36     if(s->top==-1)
37         return 1;
38     else
39         return 0;
40 }
41
42 int isFull(struct stack *s)
43 {
44     if(s->top==s->capacity-1)
45         return 1;
46     else return 0;
47 }
48
49 void push(struct stack *s,int data)
50 {
51     if(isFull(s))
52         printf("overflow");
53     else
54     {
55         s->array[++s->top]=data;
56     }
57 }
58
59 int pop(struct stack *s)
60 {
61     if(isEmpty(s))
62         printf("UnderFlow");
63     else
64     {
65         int temp= s->array[s->top];

```

```

66         s->top--;
67         return temp;
68     }
69 }
70
71 int top(struct stack *s)
72 {
73     return s->array[s->top];
74 }
75
76 // int A[5]={5,4,7,2,9,1};
77
78 int FindMaxRectInHistogram(int A[],int n)
79 {
80     struct heightindexstack * s= createHeightIndexStack();
81     int current_area, maxArea; int i;int lastindex=0;
82     for( i=0;i<n;i++)
83     {
84         if(isEmpty(s->height) || A[i]>top(s->height))
85         {
86             push(s->height,A[i]);
87             push(s->index,i);
88         }
89
90         if(A[i]<top(s->height))
91         {
92             //calculate area;
93             while(A[i]<top(s->height)&&!isEmpty(s->height))
94             {
95
96
97                 current_area=top(s->height)*(i-top(s->index));
98                 if(current_area>maxArea)
99                     maxArea=current_area;
100                 pop(s->height);
101                 lastindex=pop(s->index);
102             }
103             push(s->height,A[i]);
104             push(s->index,lastindex);
105         }
106     }
107
108     while(!isEmpty(s->height))
109     {
110         current_area=(pop(s->height)*lastindex);
111         if(current_area>maxArea)
112             maxArea=current_area;
113     }
114     return maxArea;
115 }
116
117
118 void main()
119 {
120     int A[6]={3,2,5,6,1,4,4};
121     printf("max area is %d",FindMaxRectInHistogram(A,6));
122 }

```