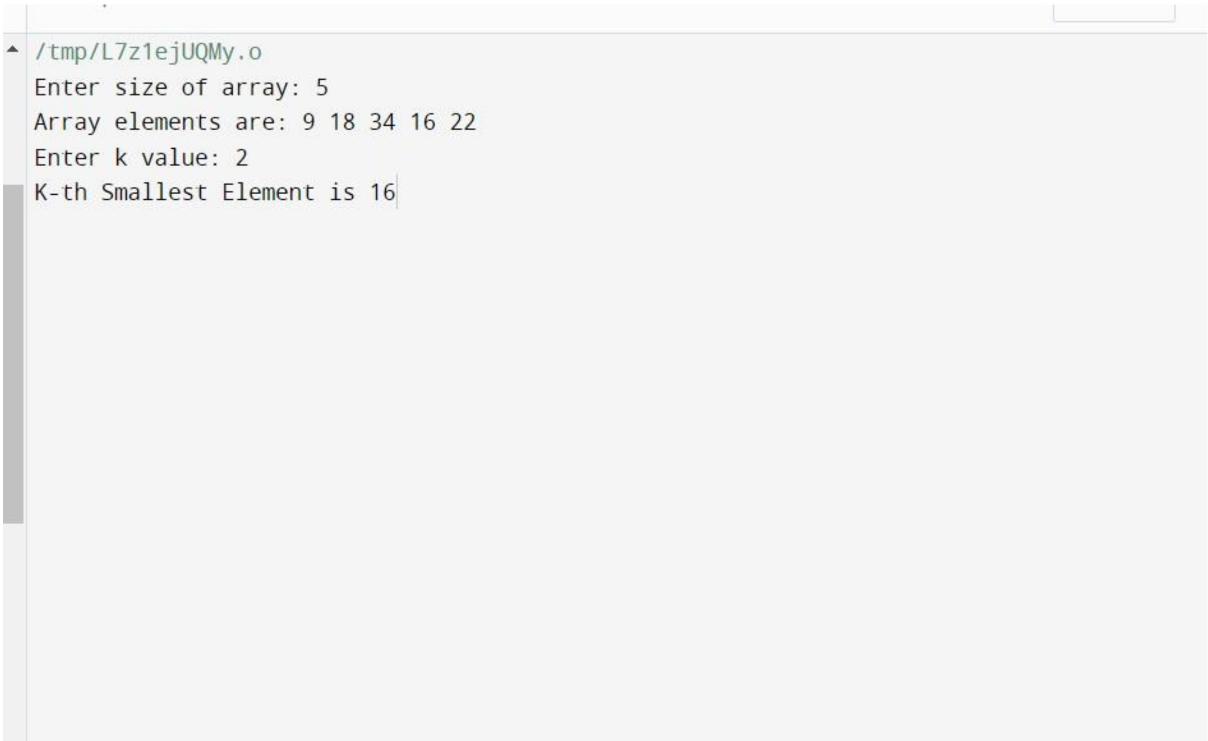
```
#include <stdio.h>

#include <stdlib.h>

typedef struct Node {

int data;

struct Node *left, *right;

} Node;

struct Node* new_node(int x)

{

struct Node* p = malloc(sizeof(struct Node));

p->data = x;

p->left = NULL;

p->right = NULL;

return p;

}

Node* insert(Node* root, int x)

{

if (root == NULL)

return new_node(x);

if (x < root->data)

root->left = insert(root->left, x);

else if (x > root->data)

root->right = insert(root->right, x);

return root;

}

int count = 0;

Node* kthSmallest(Node* root, int k)

{

// base case

if (root == NULL)

return NULL;

Node* left = kthSmallest(root->left, k);
```

```
if (left != NULL)

return left;

count++;

if (count == k)

return root;

// else search in right subtree

return kthSmallest(root->right, k);

}

void printKthSmallest(Node* root, int k)

{

Node* res = kthSmallest(root, k);

if (res == NULL)

printf("There are less than k nodes in the BST");

else

printf("K-th Smallest Element is %d", res->data);

}

int main()

{

Node* root = NULL;

int keys[20],i,keys_size;

printf("Enter size of array: ");

scanf("%d",&keys_size);

printf("Array elements are: ");

for(i=0;i<keys_size;i++){

scanf("%d",&keys[i]);

}

for (i = 0; i < keys_size; i++)

root = insert(root, keys[i]);

int k;

printf("Enter k value: ");

scanf("%d",&k);
```

```
printKthSmallest(root, k);

return 0;

}
```



```
/tmp/L7z1ejUQMy.o
Enter size of array: 5
Array elements are: 9 18 34 16 22
Enter k value: 2
K-th Smallest Element is 16
```