

## DATA STRUCTURES

1. Write a c program to implement binary search tree.

```
#include<stdio.h>

#include<stdlib.h>

struct node {

    int data;

    struct node *r;

    struct node *l;

};

void inorder(struct node* root) {

    if (root != 0) {

        inorder(root->l);

        printf("%d ", root->data);

        inorder(root->r);

    }

}

struct node* create(int x)

{

    struct node *nn;

    nn =(struct node*) malloc(sizeof(struct node));

    nn->data = x;

    nn->l = 0;

    nn->r= 0;

    return nn;

}
```

```

struct node* ins(struct node *root,int x )
{
    if(root==0)
    {
        return create(x);
    }

    if(x<root->data)
    {
        root->l=ins(root->l,x);
    }
    else
    {
        root->r=ins(root->r,x);
    }
    return root;
}

```

```

int main()
{
    struct node *root=0;

    int choice;

    do
    {
        printf("\nEnter choice:\n 1-ins 2-inorder \n");
        scanf("%d",&choice);
        switch(choice)

```

```
        {  
            case 1:  
                int x,r;  
                printf("enter data");  
                scanf("%d",&x);  
                if(root==0)  
                    root=ins(root,x);  
                else  
                    ins(root,x);  
                break;  
            case 2: inorder(root);  
                break;  
            default: printf("invalid choice");  
        }  
    }  
    while (choice!=0);  
}
```

The screenshot shows a web browser window with the URL `programiz.com/c-programming/online-compiler/`. The page title is "Online C Compiler". The main content area displays a C program in a text editor, with line numbers 50 to 70 visible. The program implements a binary search tree (BST) with two operations: insertion (1-ins) and inorder traversal (2-inorder). The output window on the right shows the execution results, including prompts for choice, data input, and the resulting inorder traversal.

```
main.c
50 printf("\nEnter choice:\n 1-ins 2-inorder \n");
51 scanf("%d",&choice);
52 switch(choice)
53 {
54     case 1:
55         int x,r;
56         printf("Enter data:");
57         scanf("%d",&x);
58         if(root==0)
59             root=ins(root,x);
60         else
61             ins(root,x);
62         break;
63     case 2: inorder(root);
64         break;
65     default: printf("Invalid choice");
66 }
67 }
68 }
69 while (choice!=0);
70 }
```

Output

```
/tmp/hJDEuTBp4h.o
Enter choice:
1-ins 2-inorder
1
Enter data:
Enter choice:
1-ins 2-inorder
1
Enter data:
Enter choice:
1-ins 2-inorder
2
4 5
Enter choice:
1-ins 2-inorder
```

2. Write a C program to implement hashing using linear probing.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define s 5
```

```
int h[s];
```

```
void insert()
```

```
{
```

```
int key,index,i,hkey;
```

```
printf("\nEnter a value to insert into hash table\n");
```

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

```
hkey=key%s;
```

```
for(i=0;i<s;i++)
```

```
{
```

```
index=(hkey+i)%s;
```

```
if(h[index] == -1)
```

```
{
```

```
    h[index]=key;
```

```
    break;
```

```
}
```

```
}
```

```
if(i == s)
```

```
    printf("\nelement cannot be inserted\n");
```

```
}
```

```
void search()
```

```
{
```

```
    int key,index,i,hkey;
```

```
    printf("\nEnter search element\n");
```

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

```
    hkey=key%s;
```

```
    for(i=0;i<s; i++)
```

```
{
```

```
    index=(hkey+i)%s;
```

```
    if(h[index]==key)
```

```
{
```

```

        printf("value is found at index %d",index);

        break;

    }

}

if(i == s)

    printf("\n value is not found\n");

}

void display()

{

    int i;

    printf("\nelements in the hash table are \n");

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

        printf("\nat index %d \t value = %d",i,h[i]);

}

int main()

{

    int j;

    for(j=0;j<s;j++)

        h[j]=-1;

    int opt,i;

    do

    {

```

```
printf("\nPress 1. Insert\t 2. Display \t3. Search \t0.Exit \n");  
scanf("%d",&opt);  
switch(opt)  
{  
    case 1:  
        insert();  
        break;  
    case 2:  
        display();  
        break;  
    case 3:  
        search();  
        break;  
    case 4:printf("invalid choice");  
}  
}while(opt!=0);  
return 0;  
}
```

```
main.c
67 int opt;
68 do
69 {
70     printf("\nPress 1. Insert\t 2. Display \t 3. Search \t 0.Exit \n");
71     scanf("%d",&opt);
72     switch(opt)
73     {
74         case 1:
75             insert();
76             break;
77         case 2:
78             display();
79             break;
80         case 3:
81             search();
82             break;
83         case 4:printf("invalid choice");
84     }
85 }while(opt!=0);
86 return 0;
87 }
```

Output

```
/tmp/entoYoGaf7.o
Press 1. Insert 2. Display 3. Search 0.Exit
1
enter a value to insert into hash table
5
Press 1. Insert 2. Display 3. Search 0.Exit
1
enter a value to insert into hash table
6
Press 1. Insert 2. Display 3. Search 0.Exit
2
elements in the hash table are

at index 0 value = 5
at index 1 value = 6
at index 2 value = -1
at index 3 value = -1
at index 4 value = -1
Press 1. Insert 2. Display 3. Search 0.Exit
```

3. Write a C program to implement bubble sort.

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int a[10],i,j,t,n;
```

```
    printf("enter size of array");
```

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

```
    printf("enter %d values\n",n);
```

```
    for(i=0;i<n;i++)
```

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

```
    for(i=1;i<n;i++) //no of passes
```

```
    {
```

```
        for(j=0;j<=n-1-i;j++) //no of comparisons
```

```
        {
```

```
            if(a[j]>a[j+1])
```

```
            {
```



```

        t=a[j];

        a[j]=a[j+1];

        a[j+1]=t;

    }

}

}

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

    printf("%4d",a[i]);

}

```

The screenshot shows a web browser window with the URL `programiz.com/c-programming/online-compiler/`. The browser has several tabs open, all labeled "Online C Compiler". The page title is "Programiz C Online Compiler". There is a "C Certification >" button in the top right corner. The main area is divided into two panels: "main.c" on the left and "Output" on the right. The "main.c" panel contains the following code:

```

1 //bubble sort
2 #include<stdio.h>
3 int main()
4 {
5     int a[10],i,j,t,n;
6     printf("enter size of array");
7     scanf("%d",&n);
8     printf("enter %d values\n",n);
9
10    for(i=0;i<n;i++)
11        scanf("%d",&a[i]);
12    for(i=1;i<n;i++) //no of passes
13    {
14        for(j=0;j<n-1-i;j++) //no of comparisons
15        {
16            if(a[j]>a[j+1])
17            {
18                t=a[j];
19                a[j]=a[j+1];
20                a[j+1]=t;

```

The "Output" panel shows the following text:

```

/tmp/Fn59THSIFQ.o
enter size of array5
enter 5 values
99 23 55 1 23
1 23 23 55 99

```

The bottom of the image shows a Windows taskbar with the search bar, task view button, and several application icons (Edge, File Explorer, VS Code, etc.). The system tray shows the temperature as 31°C, the time as 10:51 AM, and the date as 30-Aug-23.

4. Write a C program to implement insertion sort.

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int a[10],i,j,t,n;
```

```
    printf("enter size of array");
```

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

```
printf("enter %d values\n",n);
```

```
for(i=0;i<n;i++)
```

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

```
for(i=1;i<n;i++)
```

```
{
```

```
    t=a[i];
```

```
    j=i-1;
```

```
    while(j>=0 && a[j]>t) //dack traversal
```

```
    {
```

```
        a[j+1]=a[j];
```

```
        j--;
```

```
    }
```

```
    a[j+1]=t;
```

```
}
```

```
for(i=0;i<n;i++)
```

```
    printf("%4d",a[i]);
```

```
}
```

The screenshot shows a web browser with multiple tabs of 'Online C Compiler'. The active tab displays a C program for insertion sort. The code is as follows:

```
1 //insertion sort
2 #include<stdio.h>
3 int main()
4 {
5     int a[10],i,j,t,n;
6     printf("enter size of array");
7     scanf("%d",&n);
8     printf("enter %d values\n",n);
9
10    for(i=0;i<n;i++)
11        scanf("%d",&a[i]);
12    for(i=1;i<n;i++) //no of passes
13    {
14        for(j=0;j<=n-1-i;j++) //no of comparisons
15        {
16            if(a[j]>a[j+1])
17            {
18                t=a[j];
19                a[j]=a[j+1];
20                a[j+1]=t;
21            }
22        }
23    }
```

The output window on the right shows the following text:

```
/tmp/D06HZsnXGF.o
enter size of array5
enter 5 values
23 45 33 26 77
23 26 33 45 77
```

The Windows taskbar at the bottom shows the date and time as 10:51 AM on 30-Aug-23.

5. Write a c program to implement selection sort.

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int a[10],i,j,min,n,t;
```

```
    printf("enter size of array");
```

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

```
    printf("enter %d values\n",n);
```

```
    for(i=0;i<n;i++)
```

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

```
    for(i=0;i<n-1;i++)
```

```
    {
```

```
        min=i;
```

```
        for(j=i+1;j<n;j++)
```

```
        {
```

```

        if(a[j]<a[min])
        {
            min=j;
        }
    }

    if(min!=i)
    {
        t=a[i];

        a[i]=a[min];

        a[min]=t;
    }

}

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

    printf("%4d",a[i]);

}

```

The screenshot shows a web browser window with multiple tabs of 'Online C Compiler'. The active tab is 'programiz.com/c-programming/online-compiler/#google\_vignette'. The interface includes a 'Programiz C Online Compiler' header with a 'C Certification' link. Below the header, there's a code editor for 'main.c' and an 'Output' window. The code in the editor is a selection sort implementation. The output window shows the program's execution results.

```

main.c
1 //selection sort
2 #include<stdio.h>
3 int main()
4 {
5     int a[10],i,j,t,n;
6     printf("enter size of array");
7     scanf("%d",&n);
8     printf("enter %d values\n",n);
9
10    for(i=0;i<n;i++)
11        scanf("%d",&a[i]);
12    for(i=1;i<n;i++) //no of passes
13    {
14        for(j=0;j<=n-1-i;j++) //no of comparisons
15        {
16            if(a[j]>a[j+1])
17            {
18                t=a[j];
19                a[j]=a[j+1];
20                a[j+1]=t;

```

Output

```

/tmp/wqDv6J7uJG.o
enter size of array5
enter 5 values
45 32 56 20 15
15 20 32 45 56

```

6. Write a c program to implement Quick sort.

```
#include<stdio.h>
```

```
int a,ub,lb;
```

```
void swap(int *x,int *y)
```

```
{
```

```
    int t;
```

```
    t=*x;
```

```
    *x=*y;
```

```
    *y=t;
```

```
}
```

```
int partition(int a[],int lb,int ub)
```

```
{
```

```
    int pivot;
```

```
    int s,e;
```

```
    pivot=a[lb];
```

```
    s=lb;
```

```
    e=ub;
```

```
    while(s<e)
```

```
    {
```

```
        while(a[s]<=pivot)
```

```
        {
```

```
            s++;
```

```
        }
```

```
        while(a[e]>pivot)
```

```
        {
```

```
            e--;
```

```

    }

    if(s<e)
    {
        swap(&a[s],&a[e]);
    }
}

swap(&a[lb],&a[e]);

return e;
}

void quicksort(int a[],int lb,int ub)
{
    if(lb<ub)
    {
        int l=partition(a,lb,ub);

        quicksort(a,lb,l-1);

        quicksort(a,l+1,ub);
    }
}

int main()
{
    int i, a[50],n,lb,ub;

    printf("enter size of array");

    scanf("%d",&n);

    printf("enter elements\n");

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

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

```

```

        lb=0,ub=n-1;

        quicksort(a,lb,ub);

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

        printf("%d ",a[i]);

    }

```

The screenshot shows a web browser window with the URL `programiz.com/c-programming/online-compiler/`. The page title is "Online C Compiler". Below the browser window, there is a "Programiz C Online Compiler" interface. On the left, a code editor shows the following C code:

```

main.c
41 = {
42     int l=partition(a,lb,ub);
43     quicksort(a,lb,l-1);
44     quicksort(a,l+1,ub);
45 }
46 }
47 int main()
48 {
49     int i, a[50],n,lb,ub;
50     printf("enter size of array");
51     scanf("%d",&n);
52     printf("enter elements\n");
53     for(i=0;i<n;i++)
54         scanf("%d",&a[i]);
55     lb=0,ub=n-1;
56     quicksort(a,lb,ub);
57     for(i=0;i<n;i++)
58         printf("%d ",a[i]);
59
60
61 }

```

On the right, the "Output" window shows the following text:

```

/tmp/entoYoGaf7.o
enter size of array5
enter elements
10
43
56
23
99
10 23 43 56 99

```

The bottom of the image shows a Windows taskbar with various icons and a system clock indicating 9:44 PM on 13-Sep-23.

7. Write a C program to implement Merge sort

```
#include <stdio.h>
```

```
void merge(int arr[], int left, int mid, int right) {
```

```
    int i, j, k;
```

```
    int n1 = mid - left + 1;
```

```
    int n2 = right - mid;
```

```
    int L[n1], R[n2];
```

```
    for (i = 0; i < n1; i++)
```

```
        L[i] = arr[left + i];
```

```
    for (j = 0; j < n2; j++)
```

```

        R[j] = arr[mid + 1 + j];

i = 0;

j = 0;

k = left;

while (i < n1 && j < n2) {

    if (L[i] <= R[j]) {

        arr[k] = L[i];

        i++;

    } else {

        arr[k] = R[j];

        j++;

    }

    k++;

}

while (i < n1) {

    arr[k] = L[i];

    i++;

    k++;

}

while (j < n2) {

    arr[k] = R[j];

    j++;

    k++;

}

}

void mergeSort(int arr[], int left, int right) {

    if (left < right) {

```



```

        int mid = left + (right - left) / 2;

        mergeSort(arr, left, mid);

        mergeSort(arr, mid + 1, right);

        merge(arr, left, mid, right);
    }
}

int main() {

    int n;

    printf("Enter the number of elements: ");

    scanf("%d", &n);

    int arr[n];

    printf("Enter %d elements:\n", n);

    for (int i = 0; i < n; i++) {

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

    }

    mergeSort(arr, 0, n - 1);

    printf("Sorted array:\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    return 0;

}

```

Online C Compiler

programiz.com/c-programming/online-compiler/

Gmail YouTube Maps News TOP UPCOMING A... Untitled document...

Programiz C Online Compiler

C Certification >

main.c

Run

Output

Clear

```
33 }
34 }
35 void mergeSort(int arr[], int left, int right) {
36     if (left < right) {
37         int mid = left + (right - left) / 2;
38         mergeSort(arr, left, mid);
39         mergeSort(arr, mid + 1, right);
40         merge(arr, left, mid, right);
41     }
42 }
43 int main() {
44     int n;
45     printf("Enter the number of elements: ");
46     scanf("%d", &n);
47     int arr[n];
48     printf("Enter %d elements:\n", n);
49     for (int i = 0; i < n; i++) {
50         scanf("%d", &arr[i]);
51     }
52     mergeSort(arr, 0, n - 1);
53     printf("Sorted array:\n");
54     for (int i = 0; i < n; i++) {
```

/tmp/entoYoGaf7.o

Enter the number of elements: 5

Enter 5 elements:

12

76

45

32

90

Sorted array:

12 32 45 76 90

Waiting for pghbl1.pubgalaxy.com...

Type here to search

27°C

10:02 PM

13-Sep-23