LET US CONSIDER THIS Binary Search Tree:-

B

Z

C

A

Q

M

PREORDER🡪M-B-A-C-Q-Z

INORDER🡪A-B-C-M-Q-Z

POSTORDER🡪A-C-B-Z-Q-M

**5. Write a C++ program that uses functions to perform the following:**

**a) Create a binary search tree of characters.**

**b) Traverse the above Binary search tree recursively in preorder, in order and post order.**

#include<iostream>

using namespace std;

class node

{

public:

char data;

int flag;

node \*left,\*right;

};

class bint

{

public:

node \*head;

node \* create(node \*r,int d)

{

if(r == NULL)

{

r = new node;

r->data = d;

r->left = r->right = NULL;

}

else

{

if(r->data <= d)

r->right = create(r->right,d);

else

r->left = create(r->left,d);

}

return r;

}

void inorder(node \*r)

{

if(r != NULL)

{

inorder(r->left);

cout<<r->data<<" ";

inorder(r->right);

}

}

void preorder(node \*r)

{

if(r != NULL)

{

cout<<r->data<<" ";

preorder(r->left);

preorder(r->right);

}

}

void postorder(node \*r)

{

if(r != NULL)

{

postorder(r->left);

postorder(r->right);

cout<<r->data<<" ";

}

}

};

int main()

{

char d;

char ch = 'Y';

bint ob;

node \*head = NULL;

while(ch == 'Y'||ch=='y')

{

cout<<"Enter the item to insert\n";

cin>>d;

head = ob.create(head,d);

cout<<"\n Do you want to continue(y/n)";

cin>>ch;

}

cout<<"\ninorder recursive\n";

ob.inorder(head);

cout<<"\n\n";

cout<<"\npostorder rrecursive\n";

ob.postorder(head);

cout<<"\n\n";

cout<<"\npreorder recursive\n";

ob.preorder(head);

return 0;

}

**Output:-**

