Write a C program to find FIRST for predictive parser.

INPUT:

```c
#include<stdio.h>
#include<ctype.h>
void FIRST(char[],char );
void addToResultSet(char[],char);
int numOfProductions;
char productionSet[10][10];
int main()
{
    int i;
    char choice;
    char c;
    char result[20];
    printf("How many number of productions ? :");
    scanf(" %d",&numOfProductions);
    for(i=0;i<numOfProductions;i++)
    {
        printf("Enter productions Number %d : ",i+1);
        scanf(" %s",productionSet[i]);
    }
    do
    {
        printf("\n Find the FIRST of  :");
        scanf(" %c",&c);
        FIRST(result,c);
        printf("\n FIRST(%c)= { ",c);
        for(i=0;result[i]!='\0';i++)
        printf(" %c ",result[i]);
        printf("}\n");
         printf("press 'y' to continue : ");
        scanf(" %c",&choice);
    }
    while(choice=='y'||choice =='Y');
}
void FIRST(char* Result,char c)
{
    int i,j,k;
```
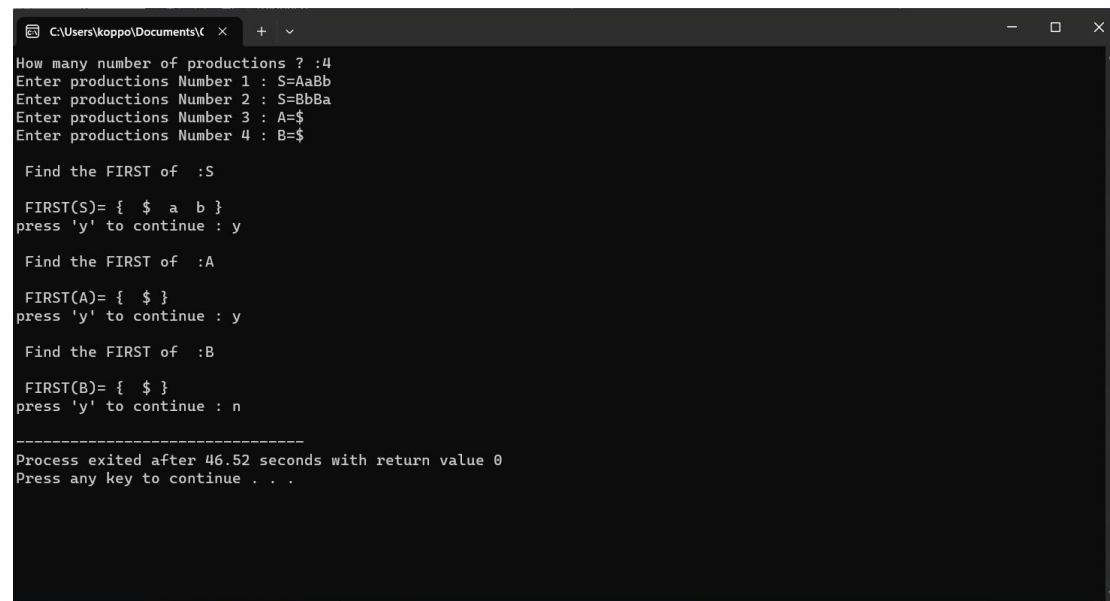
```c
   char subResult[20];
   int foundEpsilon;
   subResult[0]='\0';
   Result[0]='\0';
   if(!(isupper(c)))
   {
      addToResultSet(Result,c);
         return ;
   }
   for(i=0;i<numOfProductions;i++)
   {
      if(productionSet[i][0]==c)
      {
 if(productionSet[i][2]=='$') addToResultSet(Result,'$');
    else
        {
           j=2;
           while(productionSet[i][j]!='\0')
           {
           foundEpsilon=0;
           FIRST(subResult,productionSet[i][j]);
           for(k=0;subResult[k]!='\0';k++)
              addToResultSet(Result,subResult[k]);
            for(k=0;subResult[k]!='\0';k++)
              if(subResult[k]=='$')
              {
                 foundEpsilon=1;
                 break;
              }
            if(!foundEpsilon)
              break;
           j++;
           }
        }
      }
   }
}
   return ;
}
void addToResultSet(char Result[],char val)
{
```

```
    int k;
    for(k=0 ;Result[k]!='\0';k++)
        if(Result[k]==val)
            return;
    Result[k]=val;
    Result[k+1]='\0';
}
```

OUTPUT:

```
How many number of productions ? :4
Enter productions Number 1 : S=AaBb
Enter productions Number 2 : S=BbBa
Enter productions Number 3 : A=$
Enter productions Number 4 : B=$

 Find the FIRST of  :S

 FIRST(S)= {  $  a  b }
press 'y' to continue : y

 Find the FIRST of  :A

 FIRST(A)= {  $ }
press 'y' to continue : y

 Find the FIRST of  :B

 FIRST(B)= {  $ }
press 'y' to continue : n


--------------------------------
Process exited after 46.52 seconds with return value 0
Press any key to continue . . .
```