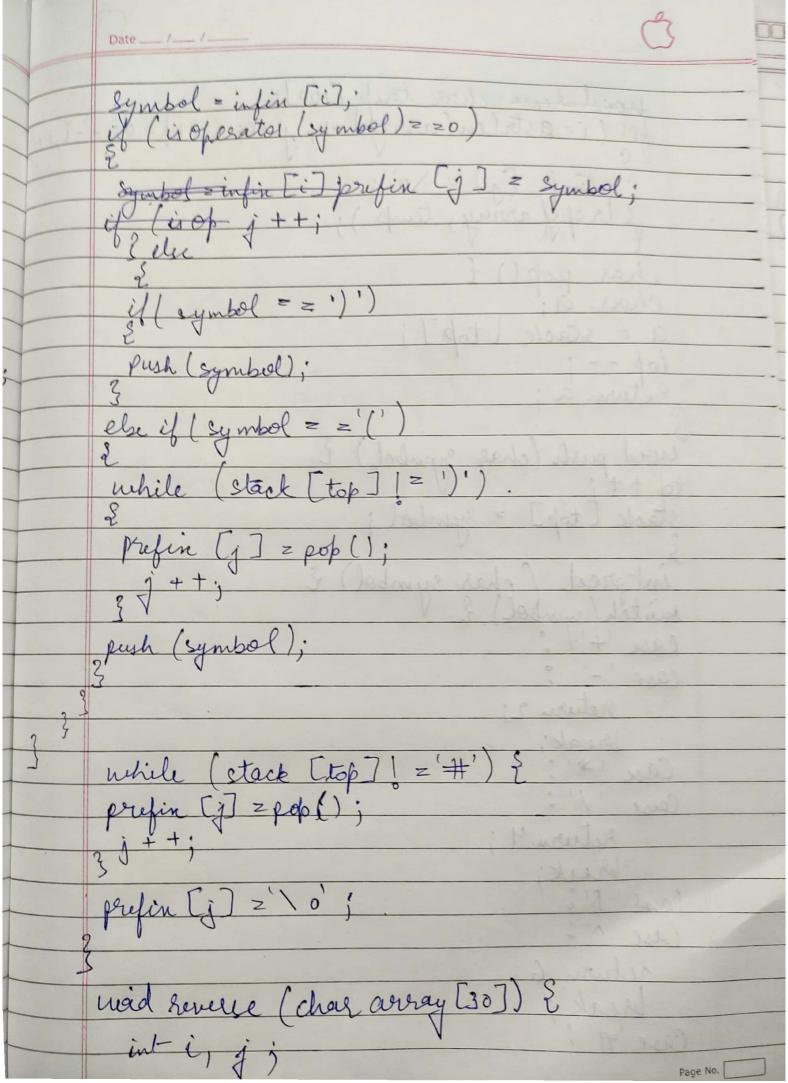
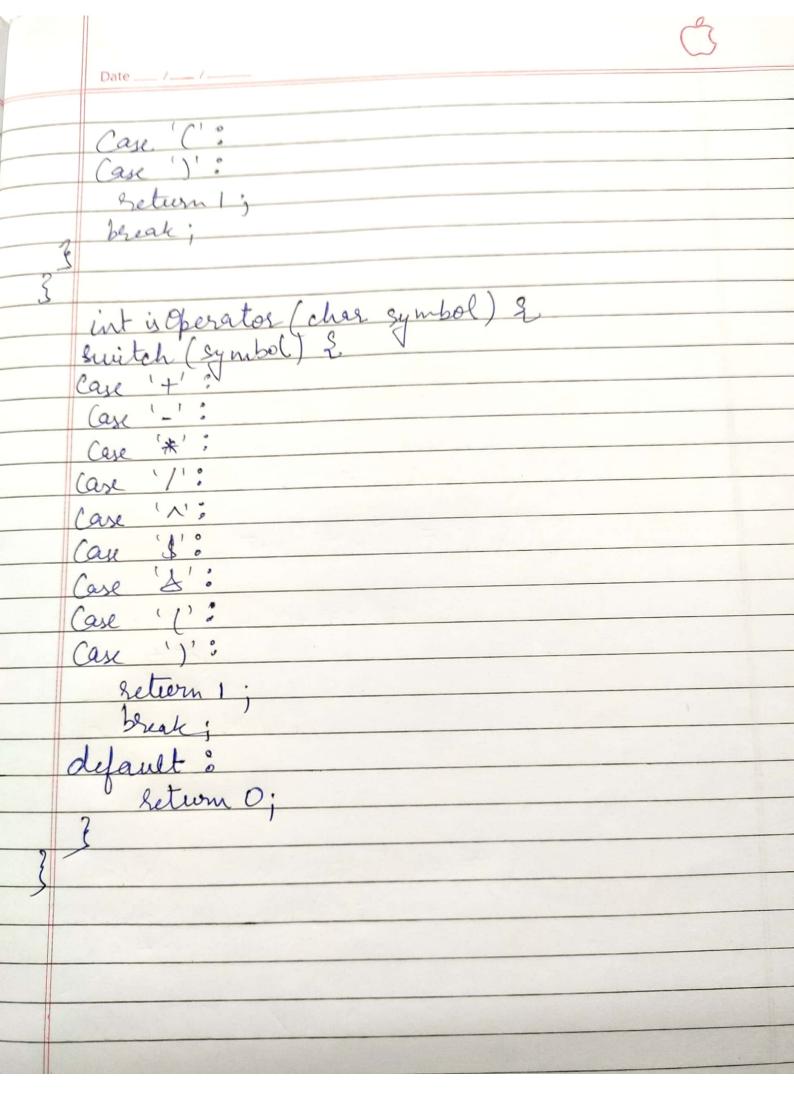
WAP to convert a given valid parantheize infin asithmetic empareion to prefix empression The empression consists of single character operands and the binary operators of (plus), - (minus), * (nultiply) Sand Mainide) # in chigle Letdio.h) # include < string. h) woid infinitoprefix (charinfin[20], char fre woid repurse (char array (30)); chas ext(): chas pop (); usid push (cher symbol); aint top = -1; Char stack [MAX]; main ().2 Chair infine [20], prefin[20], temp;
Print ("Enter infin operation: "). gets infin!; Shfin toprefin (infin, prefin); Relieve (prefin); noid infintoprefix (char infix tro], charprefix to inti, 1 = 0) chas symbol; stack [4+ top] = '#'; senerse (infin); for (i=0; (< storlen (infin); i++) }



```
troid sense chois temp [100]; i+il=0;-in

for (i= Ostalen (array)-1, f

S
     temp [j] = '\o';
trepy array, temp);
    = stack [top];
top + +;
intpred (char symbol) ?
suitch (symbol) ?
case '+'V.
Case (*);
  return 4;
  break;
  Return 6;
break;
```



postfix on hours # include (stdio.h) clouble compute (Schor symbol, double ops, double) # include <ctype.h> switch(symbol) : return 6/1 + 0/2; Cax '*': return op! * op2; Cax '*': return op! * op2; Cax '/': return op! / op2; Case \$': Case (2); Seturn pow (opi, op2); noid main (ouble S[20]; uble op, op2; Charpoilfin [20], Symbol; Brintf (2 Later postfix empression); Scanf ("1.5", postfix); top = -1;
et (i=0; i < stelen (postfine); i ++)

Date __ /__ /_ synbol = postfix [i];

if (is digit (symbol))

S[++top] = symbol-'o'; of2 = s[top --]; of 1 = S[top--]; see = compute (symbol, of, op2 S[++ top] = ses,

4) WAP to perform CsCD of two number using Recursion. # include (stolio. h) int GCD (int, int); int main() int num 1, num 2 see;

printf ("\n Enter the two numbers:");

printf ("\n', d',d", & num!, & num!);

ses = GCD Enum! num 2);

printf ("\n GCD) 1.d and 1.d > 1.d", num!,

printf ("\n GCD);

rum 2, res); int GCD (int n, inty) int rem; (GCD (y, run));