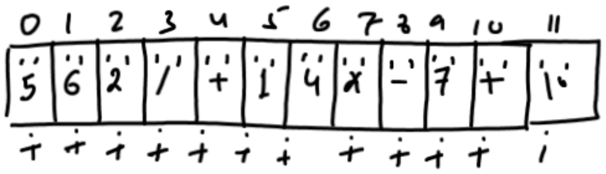
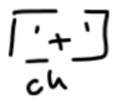
Implementing Evaluation Of Postfix Expression Algorithm

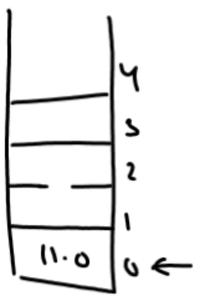
```
struct Stack
{
    float arr[10];
    int tos;
};
void push(struct Stack *,float);
float pop(struct Stack *);
int isoperand(char);
float calculate(float,float,char);
float evaluate(char[ ]);

int main()
{
   char postfix[20];
   float ans;
   printf("enter a valid postfix exp:");
   scanf("%s",postfix);
   ans=evaluate(postfix);
   printf("Result is %f",ans);
   return 0;
}
```

```
float evaluate(char postfix[20])
{
    struct Stack S;
    int i;
    char ch;
    float op1,op2,res;
    S.tos=-1;
    for(i=0;postfix[i]!='\0';i++)
    {
        ch=postfix[i];
        if(isoperand(ch)==1)
            push(&S,ch-48);
                        50 - 48
        else
        {
            op2=pop(&S); 7.0
            op1=pop(&S); 4.0
            res=calculate(op1,op2,ch);
            push(&S,res); 4.0, 7.0, '+'
        }
    }
    res=pop(&S);
    return res;
}
```







```
int isoperand(char ch)
{
    if(ch>=48 && ch<=57)
            return 1;
    else
            return 0;
}
OR
int isoperand(char ch)
{
    return(ch>=48 && ch<=57);

}
```

```
float calculate(float op1,float op2,char ch)
{
    switch(ch)
    {
        case '+':
                    return op1+op2;
        case '-':
                     return op1-op2;
        case '*':
                    return op1*op2;
        case '/':
                    return op1/op2;
         case '%':
                    return fmod(op1,op2);
         case '$':
                    return pow(op1,op2);
         default:
                     return 0.0;
     }
}
```

ASSIGNMENT
===========
WAP to evaluate PREFIX expression given by the user.