① Write C program to evaluate the valid prefix expression
② Write a program to convert from infix to prefix
③ Recursive function to find factorial

①:
```
#include <stdio.h>
#include <string.h>
#include <math.h>
double compute (char symbol, double op1, double op2)
{ switch (symbol)
  { case '+' : return op1+op2;
    case '-' : return op1-op2;
    case '*' : return op1*op2;
    case '/' : return op1/op2;
    case '$' :
    case '^' : return pow(op1,op2);
  }
}
void main()
{ double s[20];
  double op1, op2, res;
  int top, i;
  char postfix[20], symbol;
  top = -1;
  printf("Enter postfix exp: ");
  scanf("%s", &postfix);
  { symbol = postfix[i];
    if (isdigit(symbol))
    { s[++top] = symbol - '0';
    }
    else
    { op2 = s[top--];
      op1 = s[top--];
      res = compute(symbol, op1, op2);
      s[++top] = res;
    }
  }
```

```
        res = s[top--];
        printf("Result : %f", res);
    }
```

(2)
```
#include <stdio.h>
#include <string.h>
int F (char symbol)
{   switch (symbol)
    {   case '+':
        case '-': return 1;
        case '/':
        case '*': return 3;
        case '^':
        case '$': return 6;
        case '(': return 0;
        case '#': return -1;
        default : return 8;
    }
}
int G(char symbol)
{   switch (symbol)
    {   case '+':
        case '-': return 2
        case '/':
        case '*': return 4;
        case '^':
        case '$': return 5;
        case '(': return 9;
        case ')': return 0;
        default : return 7;
    }
}
```

```
void infix_prefix (char infix[7], char prefix[7])
{   int top=-1, j=0, i;
    char s[30];
    char symbol;
    s[++top] = '#';
    strrev(infix);
    for (i=0; i<strlen(infix); i++)
    {   symbol = infix[i];
        while (F(s[top]) > G(symbol))
        {   prefix[j] = s[top--];
            j++;
        }
        if (F(s[top]) != symbol)
            s[++top] = symbol;
        else
            top--;
    }
    while (s[top] != '#')
    {   prefix[j++] = s[top--];
    }
    prefix[j] = '\0';
    strrev(prefix);
    printf("prefix : %s", prefix);
}
void main()
{   char infix[20];
    char prefix[20];
    printf("Enter infix:")
    scanf("%s", &infix);
    infix_prefix(infix, prefix);
}
```

(9.)

```
#include <stdio.h>
int fact (int n)
{   if (n==0) return 1;
    return n * fact (n-1);
}
void main()
{   int n;
    printf ("Enter a number: ");
    scanf ("%d", &n);
    printf ("Factorial : %d", fact(n));
}
```