Q1:- (1)

```
#include <iostream>
using namespace std;
node * reverse (node *head)
{
  node* p = head;
  node * p2 = p->next
  *p -> next = NULL;
   p -> prev = p2;
while (p2!=NULL)
{
   p2 -> prev = p2 -> next;
   p2 -> next = p;
   p = p2;
   p2 = p2 -> prev;
}
   head = p;
   return head;
}
```

Q2:-

```
#include <stack>
char stack[20];
int top=-1
Void Push(char x)
{
  stack[++top]=x;
}
Char Pop()
{
    if(top==-1)
    {return -1;
    }
  else
    {
      return stack[top--];
    }
}
int priority(char x)
{ if(x=='(')
   {return 0;
   }
else if(x=='+'||x=='-')
     {return 1;
     }
else if(x=='*'||x=='/')
     {return 2;
     }
}
```

```
int main ()
{
  char exp[20]
  char e, x;
  cout << "Enter the expression "<<endl;
  cin>> exp;
  e = exp;
While (e! = '\0')
{
  if (is alum (e))
     cout << e;
  else if (e = = '(')
        Push (e);
  else if (e = = ')')
  { while ((x = pop()) != '(')
        cout << x;
  }
   else
   {
     while (Priority (stack [top]) >= priority (e))
      { cout <<pop();
      }
      Push (e)
     }
        e++;
```

```
while (top != -1)
{
    cout << pop();
}

return 0;
}
```