```
Write a C++ program using stack ADT to balance the braces.

Parenthesis and brackets.

Header file:

# include \( \) iostream >

# include \( \) string >

using namespace std;

class AStack {\( \)

Int capacity;
```

int top:

when "arm;

public:

Astack();

Astack();

VAStack();

Void punh(chan ell);

Void pop();

Andropeek();

Lool is Empty();

bool is Balanced (storing expn):

Implementation file:

3:

# include "head.h";

AStack:: Astack() {

dop:-1;

capacity: 10;

avr:= new char[capacity]:

3

```
A Stack: : A Stack C int size of
                top = -1;
                capacity: size;
                over = new chan [ capacity];
        3
    AStack : : ~ AStack (){
              delete [] wor;
void A Stack: puch (char ele) {
            if ( top 1 capacity -1){
                  top++;
                 arritop] = ele;
              else &
                   couts 11 "Stack is full";
     3
   void AStack :: pop(){
             if (top! =-1) s
              else f
                 cout 11 " Stack is empty";
   bool Astack: is Empty Uf
              oreturn ( top = = -1);
   chan
         AStack :: puek() {
              if ( top != -1) {
                    return aron [top]: 3
```

```
celse &
               cout 11 "Stack is empty";
            3
bool
     is Matching Cahan open, whan closes?
            if copen== 'C' xx close == ')') return true;
           if (span == 'l' dd close == 'z') return tome:
           if (open == '[' aa dose == ']') return tome:
           return false;
        is Balanced Cotting export
bool
          AStack stack (expn. length(1);
          for (int i = 0: iz expresength (); i++ );
                 chan ch = exposis;
                  if C ch == 'C' 11 ch == 'E' 11 ch == '['){
                         Stack push (1);
                  else if (ch == ')' 11 ch = '3' 11 ch == 'J') {
                      if Custack is Empty Will is Matching (stack needs
                                                   eh)) E
                             return false;
                      stack. popls
                oreturn stack is Empty ();
```

```
Main file:
           import "header.h";
               main () &
           String stri
           cout 12 "Enter the expression: ";
           cin >> stn;
           if (is Balanced (stor) {
                 cout 22 "Expression is balanced.";
             else ¿
                 Cout LL " Expression is not balanced.".
Write a C++ program using Stack ADT to convert
 infix to postfix expussion:
Program:
Headen file:
       # include 2 iostoream >
       # include 2 string>
      susing mames pace std;
       class Astack [
                  int capacity:
                 int dop;
                char & wor;
                public:
                    A Stack ();
                    Astack (int size);
                   ~Astack();
                   void push ();
```

```
cout 12 "Stack is empty";
            3
bool
     is Matching Cuhan span, whan closes?
            if (open == 'C' xx close == ')') outurn true;
           if ( spen == 'E' & & close == '3' > outson true:
           if (open == '[' aa dose == ']') return true:
           return false;
 tool
         is Balanced Cotting export
           AStack stack (expn. length());
           for (int i = 0: i / expresength (); i++ );
                chan (h = exp [i];
                  if C wh == 'C' 11 ch == 'E' 11 ch == '['){
                         Stack push (9)
                else if (ch == ')' 11 ch = ']' 11 ch == ']') {
                      if Custack is Empty Will is Matching Cotack puts
                                                   uch)){
                              retion false;
                       stack popls
                 oreturn stack is Empty();
```

else &

whom peck(); dood is Empty (); unt precedence ( whom op); String infix to Postfix Cstring expans Implement file: # include "head h" A Stack :: Astack () { top = -1; Capacity = 10: aun: nw char[capacity]; AStack: : Astack (int size) { top = - 1: capacity: size: aron = new char[ capacity]; } A Stack :: ~ A Stack () } idelete[] wor; 4 void AStack: puch (chan ale); if (top z capacity-1) { top++; avor [top] = ele: else s cout 12" Stack is full"; } void Astack: upop() & if (top!= -1) & top -- ; 3 elses

(out 11 "Stack is ampty"; }

void push ( char ele);

```
whan Astack :: pulk () of
          if (top! =-1) {
                 outurn ausstop]: 3
           else s
                cout 11 "Stack is ampty" : 3
 bool AStack: is Empty() [
           outurn (top == -1):
 unt greadence ( than op) {
          if ( op = = (+ , 11 op = = 1-, )
                 outwon 1;
           if ( op == 1 * 1 | 1 | op == 1/1) {
                oretron 2: 3
           oretwin o:
   3
string
        infix to Postfix ( string exp) &
         AStack stack (exp. length (1);
         String p = "";
         for ( ind i=0; i 2 exp. length(): j++)&
               whom ch = exptil's
                if (isalnum (chs) {
                        p+= ch; 4
                else if ( th = = 1(') &
                      Statch. push (ch ); }
                else if (ch = = ')'){
                     while ( ! stack is Empty () at stack pek() !=
                                                        1(,7)
                           b # = stack . pop();
```

stack pop W:

```
else f
                   while ( Stack is Empty () as preadence ( stack peck ()
                                    > = gouadence (ch)){
                            p + = stack . pop ();
                    stack. puch (ch);
             3
             while (! stack · is Empty 1)) [
                       p+= stack pop();
           return p:
       hypor to be made house too with met
Main file:
        #indude "head.h";
        int main() {
              string exp:
              cout 12 " Enter con infix expossion: "3
              in >> exp;
               isting p = infix to Postfix ( exp);
               cout 12 " Postfix expression: " 21 p 21 endl;
               oretwin o;
```

Output:

Enter an infix expousion: a+b\*g-k7. n+m

Postfix expousion: abg \*+k-nm+9.

Output for Iralance Stack ADt program:

Enter the expression: Praye Expression is not tralanced.

Textrase 2:

Expussion is balanced.

Write a C++ program using List ADT- Annay implementation to salue the for egeneral values of M and N. What is the trunning time of the program?

Header Lile:

#indude ziorbeams;

using mamerpaad std;

idas List?

int sliments;

int succentsize;

public:

List();

List();

void eliminata(int index);

int get (intindex);

int getsize();

```
Implementation file:
  # include " head h "
  List():: List() {
             size = 0:
            capacity = 15;
             avor = new int [ capacity ] : }
    List(): List(int c, int s, int *a)
               canacity = (;
               size = A;
               our = new int [capacity]:
              for Cint 1=0: 12 size: (+) (
                       amrij = a [i];
                3
  Lint():: ~ Lint() &
               capacity = 0:
                size = 0;
                delete [] avn; z
  void List(): eliminata (int indux)
             for (int i = index; i a conventsize - 1; i++ ) f
                           artifij: artifiti):
               wountsize -- ; 7
           List :: get Cindex)
    int
                   orekunn con [index]; ]
    int
              :: getsize () {
                    return correntsize; 3
```

```
Main file:
        #include "head h";
        ind main() {
               int N.M:
               int capacity;
               cout 12 " Enter the munter of people (N): ";
               an ss N:
               cout 22 "Enter the number of passes (M): ";
               an ssM:
                int convert Index = 0;
                while (people getsize () > ) {
                       convert Index = ( covert Index + N-1) %
                                               people. getSize();
                       people eliminate ( avount Index):
                couter "The winner is person: " 22 people get(0)
                                                  LLendl;
                outurn o;
Output:
Testrase inter the number of people (N): 89
        Enter the number of paines (M): 70
         The winner is person: 28
Test case 2:
         Enter the number of people (N): 60
         Enter the mumber of passes (M): 50
```

The winner is person: 51