```
%{
#include <bits/stdc++.h>
using namespace std;

extern string yytext;
extern int yylex();
extern int yyparse();
extern FILE *yyin,*yyout;
void yyerror(const char *s);

#define pi acos(-1)
bool isprime[100005];
vector<int>pr;
int currentex = 0;
int ifelfound = 0;
void siv()
{
pr.push_back(2);
for(int i=3;i*i<=100000;i+=2)
    if(isprime[i]==0)
        for(int j=i*i;j<=100000;j+=2*i) isprime[j]=1;
for(int i=3;i<=100000;i+=2)
    if(isprime[i]==0) pr.push_back(i);
}

int bigmod(int a,int b,int c){
    if(b==0) return 1;
    if(b==1) return a%c;
    int x=bigmod(a,b/2,c)%c;
    x=(x*x)%c;
    if(b%2) return (a*x)%c;
    else return x;
}
map<string,int>varmap;
map<string,int>ivar;
map<string,vector<int> >aivar;
map<string,double>dvar;
map<string,vector<double> >advar;
map<string,char>cvar;
map<string,vector<char> >acvar;
map<string,bool>bvar;
map<string,vector<bool> >abvar;
void create_array(string name,int size,int type)
{
    if(type == 6){
        for(int i = 0 ; i < size ; i++)
            aivar[name].push_back(0);
    }
    else if(type == 8){
        for(int i = 0 ; i < size ; i++)
            advar[name].push_back(0.0);
    }
    else if(type == 9){
        for(int i = 0 ; i < size ; i++)
            acvar[name].push_back('#');
    }
    else if(type == 10){
        for(int i = 0 ; i < size ; i++)
            abvar[name].push_back(false);
    }
}

%}

%union {
  int ival;
```

```
 67    char cval;
 68    double fval;
 69    bool bval;
 70    char *type,*name;
 71    char *str;
 72    char c;
 73  }
 74
 75  %union {
 76    struct{
 77      int ival;
 78      double fval;
 79      char cval;
 80      bool bval;
 81      char *str;
 82    }vars;
 83  }
 84
 85  %token HASH SHOW READ MCMNT SCMNT FUNC RETURN FUNCEND QUOTE OPENBR1 CLOSEBR1 OPENBR3 CLOSEBR3 MAIN INCLUDE
GREATER LESS GREATEREQ LESSEQ MOD ASIN ACOS ATAN SIN COS TAN SQRT POW BIGM FIBO SQ PRIME ISPRM QUBE GCD LCM DOT
PLUS MINUS MUL DIV SEMI COMM EQUAL IF ELSE ELIF ENDIF END LOOP BY IN ENDLOOP WHILE ENDWHILE AND OR NOT COLON
 86  %token NEWLINE
 87  %token <str> STRING
 88  %token <str> HEADFILENAME
 89  %token <type> INT
 90  %token <type> CHAR
 91  %token <type> DOUBLE
 92  %token <type> BOOL
 93  %token <c> PERCENT
 94  %token <name> VARNAME
 95  %token <ival> DIGIT
 96  %token <fval> FLOATVAL
 97  %token <cval> CHARACTER
 98  %token <bval> TRUE
 99  %token <bval> FALSE
100  %token <bval> BOOLVAL
101  %type <vars> VARIABLE
102  %type <vars> VALUE
103  %type <type> TYPE
104  %type <vars> EX
105  %type <vars> F
106  %type <vars> T
107  %type <bval> condx
108  %type <bval> CONDX
109  %type <vars> A
110  %type <fval> B
111  %type <fval> C
112  %type <ival> NUMVALUE
113  %type <fval> else
114  %type <fval> NUMV
115  %type <fval> EXP
116  %type <fval> IFELSE
117  %start START
118
119  %%
120  START: hdrs funcs MAIN LINE END     {  cout << "//Main function executed successfully" << endl; }
121  ;
122  hdrs:
123           |hdrs header
124  ;
125  header:     HASH INCLUDE LESS HEADFILENAME GREATER      { cout << "//Header file " << $4 << " included" <<
endl; }
126  ;
127  LINE :
128       |LINE ST
129  ;
```

```
130
131  ST : read
132       |show           {cout<<endl;}
133       |VD             { cout << endl; }
134       |EXP            { cout << "Result of expression = " << $1 << endl; }
135       |condx
136       |IFELSE         { cout << "//Value of if-else block : " << $1 << endl; }
137       |FL
138       |WH
139       |empty
140       |singlecomment
141       |multicomment
142       |fcall
143  ;
144  VD: TYPE VARNAME EQUAL VALUE SEMI       { if( varmap.find($2) == varmap.end() ){
145                            if(strcmp($1,"int") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with " << $4.ival ; varmap[$2] = 1; ivar[$2] = $4.ival; }
146                            else if(strcmp($1,"double") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.fval ; varmap[$2] = 3; dvar[$2] = $4.fval; }
147                            else if(strcmp($1,"char") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.cval ; varmap[$2] = 4; cvar[$2] = $4.cval;  }
148                            else if(strcmp($1,"bool") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.bval ; varmap[$2] = 5; bvar[$2] = $4.bval; }
149                            }
150                            else { yyerror("Error : Variable "); yyerror($2); yyerror(" previously declared");
exit(-1); }
151                            }
152       |TYPE VARNAME EQUAL VARIABLE SEMI   { if( varmap.find($2) == varmap.end() )
153                            {
154                            if(strcmp($1,"int") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.ival ; varmap[$2] = 1; ivar[$2] = $4.ival;  }
155                            else if(strcmp($1,"double") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.fval ; varmap[$2] = 3; dvar[$2] = $4.fval; }
156                            else if(strcmp($1,"char") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.cval ; varmap[$2] = 4; cvar[$2] = $4.cval; }
157                            else if(strcmp($1,"bool") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.bval ; varmap[$2] = 5; bvar[$2] = $4.bval; }
158                            }
159                            else { yyerror("Error : Variable "); yyerror($2); yyerror(" previously declared");
exit(-1);}
160                            }
161       |TYPE VARNAME EQUAL EX SEMI     { if( varmap.find($2) == varmap.end() ){
162                            if(strcmp($1,"int") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.ival ; varmap[$2] = 1; ivar[$2] = $4.ival; }
163                            else if(strcmp($1,"double") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.fval ; varmap[$2] = 3; dvar[$2] = $4.fval; }
164                            else if(strcmp($1,"char") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.cval ; varmap[$2] = 4; cvar[$2] = $4.cval; }
165                            else if(strcmp($1,"bool") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< $4.bval ; varmap[$2] = 5; bvar[$2] = $4.bval; }
166                            }
167                            else { yyerror("Error : Variable "); yyerror($2); yyerror(" previously declared");
exit(-1); }
168                            }
169       |TYPE VARNAME SEMI          { if( varmap.find($2) == varmap.end() ){
170                            if(strcmp($1,"int") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< 0 ; varmap[$2] = 1; ivar[$2] = 0; }
171                            else if(strcmp($1,"double") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with "<< 0.0 ; varmap[$2] = 3; dvar[$2] = 0.0; }
172                            else if(strcmp($1,"char") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with #" ; varmap[$2] = 4; cvar[$2] = '#';}
173                            else if(strcmp($1,"bool") == 0) {cout << "//"<<$1<<" Variable " << $2  << " is
initialized with false" ; varmap[$2] = 5; bvar[$2] = false; }
174                            }
175                            else { yyerror("Error : Variable "); yyerror($2); yyerror(" previously declared");
exit(-1); }
```

```
176                         }
177      |TYPE VARNAME OPENBR3 DIGIT CLOSEBR3 SEMI { if( varmap.find($2) == varmap.end() ){
178                             if(strcmp($1,"int") == 0) { cout <<"//"<<$1<<" Array " << $2 << " of size "
<<$4<<" is declared." ; varmap[$2] = 6;  create_array($2,$4,6); }
179                             else if(strcmp($1,"double") == 0) { cout <<"//"<<$1<<" Array " << $2 << " of
size " <<$4<<" is declared." ; varmap[$2] = 8;  create_array($2,$4,8);}
180                             else if(strcmp($1,"char") == 0) { cout <<"//"<<$1<<" Array " << $2 << " of size
" <<$4<<" is declared." ; varmap[$2] = 9;  create_array($2,$4,9);}
181                             else if(strcmp($1,"bool") == 0) { cout <<"//"<<$1<<" Array " << $2 << " of size
" <<$4<<" is declared." ; varmap[$2] = 10; create_array($2,$4,10);}
182                             }
183                             else { yyerror("Error : Variable "); yyerror($2); yyerror(" previously
declared");  exit(-1); }
184                             }
185  ;
186  VALUE:  DIGIT              {$$.ival = $1; currentex = 1}
187          |FLOATVAL          {$$.fval = $1; currentex = 2}
188          |CHARACTER         {$$.cval = $1; currentex = 4}
189          |BOOLVAL           {$$.bval = $1; currentex = 5}
190          |STRING            {$$.str = $1;  currentex = 0;}
191          |VARNAME OPENBR3 NUMVALUE CLOSEBR3  { if(varmap.find($1) == varmap.end()) { yyerror("Error :
Variable "); yyerror($1); yyerror(" not declared\n"); exit(-1);}
192                             else {
193                                 if(varmap[$1] == 6) { if( $3 <= aivar[$1].size()){$$.ival = aivar[$1][$3];}
else {yyerror("Error : Array index out of range");} }
194                                 else if(varmap[$1] == 8) { if( $3 <= advar[$1].size()){$$.fval =
advar[$1][$3];} else {yyerror("Error : Array index out of range");} }
195                                 else if(varmap[$1] == 9) { if( $3 <= acvar[$1].size()){$$.cval =
acvar[$1][$3];} else {yyerror("Error : Array index out of range");} }
196                                 else if(varmap[$1] == 10) { if( $3 <= advar[$1].size()){$$.bval =
abvar[$1][$3];} else {yyerror("Error : Array index out of range");} }
197                             }}
198  ;
199
200  read:       READ OPENBR1 VARNAME CLOSEBR1 SEMI      {
201                                     if(varmap.find($3) == varmap.end()) { yyerror("Error : Variable ");
cout << $3 ; yyerror(" not declared\n"); exit(-1);}
202                                     else {
203                                         if(varmap[$3] == 1) { if(currentex == 1) { int a; scanf("%d",&a);
ivar[$3] = a;cout <<"Value of "<<$3<< " read as " <<a<< endl;}  }
204                                         else if(varmap[$3] == 3) { if(currentex == 2) {double a;
scanf("lf",&a); dvar[$3] = a;cout <<"Value of "<<$3<<" read as "<<a<<endl; } }
205                                         else if(varmap[$3] == 4) { if(currentex == 4) {char a;
scanf("%c",&a); cvar[$3] = a;cout <<"Value of "<<$3<<" read as "<<a<< endl; } }
206                                     }
207                                     }
208          |READ OPENBR1 VARNAME OPENBR3 NUMVALUE CLOSEBR3 CLOSEBR1 SEMI       {
209                             if(varmap.find($3) == varmap.end()) { yyerror("Error : Variable ");
yyerror($3); yyerror(" not declared\n"); exit(-1);}
210                             else {
211                                 if(varmap[$3] == 6) { if( $5 <= aivar[$3].size()){  int a; scanf("%d",&a);
aivar[$3][$5] = a; cout << "Value of " << $3 << "[" << $5 << "]" << " read as " << a << endl;} else
{yyerror("Error : Array index out of range");} }
212                                 else if(varmap[$3] == 8) { if( $5 <= advar[$3].size()){ double a;
scanf("lf",&a); advar[$3][$5] = a; cout << "Value of " << $3 << "[" << $5 << "]" << " read as " << a << endl;}
else {yyerror("Error : Array index out of range");} }
213                                 else if(varmap[$3] == 9) { if( $5 <= acvar[$3].size()){ char a;
scanf("%c",&a); acvar[$3][$5] =  a; cout << "Value of " << $3 << "[" << $5 << "]" << " read as " << a << endl;}
else {yyerror("Error : Array index out of range");} }
214                             }}
215  ;
216  show:       SHOW OPENBR1 SHOWSTR CLOSEBR1 SEMI {cout << endl; }
217  ;
218  SHOWSTR:
219          |STRING         { cout << $1 ; }
220          |VARNAME        { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable "); yyerror($1);
```

```
yyerror(" not declared\n"); exit(-1);}
                        else {
                          if(varmap[$1] == 1) { cout << ivar[$1]; }
                          else if(varmap[$1] == 3) { cout << dvar[$1]; }
                          else if(varmap[$1] == 4) { cout << cvar[$1];  }
                          else if(varmap[$1] == 5) { cout << bvar[$1]; }
                       }
                       }
         |VARNAME OPENBR3 NUMVALUE CLOSEBR3  { if(varmap.find($1) == varmap.end()) { yyerror("Error :
Variable "); yyerror($1); yyerror(" not declared\n"); exit(-1);}
                              else {
                                  if(varmap[$1] == 6) { if( $3 <= aivar[$1].size()){cout << aivar[$1][$3];}
else {yyerror("Error : Array index out of range");} }
                                  else if(varmap[$1] == 8) { if( $3 <= advar[$1].size()){cout <<
advar[$1][$3];} else {yyerror("Error : Array index out of range");} }
                                  else if(varmap[$1] == 9) { if( $3 <= acvar[$1].size()){cout <<
acvar[$1][$3];} else {yyerror("Error : Array index out of range");} }
                                  else if(varmap[$1] == 10) { if( $3 <= advar[$1].size()){cout <<
abvar[$1][$3];} else {yyerror("Error : Array index out of range");} }
                              }}
 ;
 VARIABLE:   VARNAME { if(varmap.find($1) == varmap.end()) { yyerror("Error  : Variable "); yyerror($1);
yyerror(" not declared\n"); exit(-1);}
               else {
              if(varmap[$1] == 1) { cout << ivar[$1] << endl; $$.ival = ivar[$1]; }
               else if(varmap[$1] == 3) { $$.fval = dvar[$1]; }
               else if(varmap[$1] == 4) { $$.cval = cvar[$1];  }
                   else if(varmap[$1] == 5) { $$.bval = bvar[$1]; }
             }
         }
 ;
 TYPE: INT       {$$ = $1;}
     |CHAR       {$$ = $1;}
     |DOUBLE     {$$ = $1;}
     |BOOL       {$$ = $1;}
 ;

 EXP:    VARNAME EQUAL EX SEMI       { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable ");
yyerror($1); yyerror(" not declared\n"); exit(-1);}
                      else {
                          if(varmap[$1] == 1) { currentex = 1; ivar[$1] = $3.ival; $$ = (double)($3.ival); }
                          else if(varmap[$1] == 3) { currentex = 2; dvar[$1] = $3.fval; $$ = $3.fval; }
                      }
                      }
 ;

 EX: EX PLUS T           { if(currentex == 1) { $$.ival = $1.ival + $3.ival; } else { $$.fval = $1.fval +
$3.fval; } }
     |EX MINUS T         { if(currentex == 1) { $$.ival = $1.ival - $3.ival; } else { $$.fval = $1.fval -
$3.fval; } }
     |T              { if(currentex == 1) { $$.ival = $1.ival ; } else { $$.fval = $1.fval ; } }
 ;
 T:  T MUL F         { if(currentex == 1) { $$.ival = $1.ival * $3.ival; } else { $$.fval = $1.fval *
$3.fval ; } }
     |T DIV F            { if(currentex == 1) { $$.ival = $1.ival / $3.ival ;} else { $$.fval = $1.fval /
$3.fval;  } }
     |F              { if(currentex == 1) { $$.ival = $1.ival ; } else { $$.fval = $1.fval ; } }
 ;
 F:   OPENBR1 EX CLOSEBR1        { if(currentex == 1) { $$.ival = $2.ival; } else { $$.fval = $2.fval; } }

     |VARNAME            { if(varmap.find($1) == varmap.end()) { yyerror("Error  : Variable "); yyerror($1);
yyerror(" not declared\n"); }
                        else {
                          if(varmap[$1] == 1) { $$.ival = ivar[$1]; currentex = 1;}
                          else if(varmap[$1] == 3) { $$.fval = dvar[$1]; currentex = 3;}
                          else { yyerror("Error : Expected an integer or double"); }
```

```
273                       }
274                     }
275      |DIGIT              { $$.ival = $1; currentex = 1;}
276      |FLOATVAL           { $$.fval = $1; currentex = 2; }
277      |VARNAME OPENBR3 NUMVALUE CLOSEBR3  { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable
"); yyerror($1); yyerror(" not declared\n"); exit(-1);}
278                                 else {
279                                     if(varmap[$1] == 6) { if( $3 <= aivar[$1].size()){$$.ival = aivar[$1][$3];}
else {yyerror("Error : Array index out of range");} }
280                                     else if(varmap[$1] == 8) { if( $3 <= advar[$1].size()){$$.fval =
advar[$1][$3];} else {yyerror("Error : Array index out of range");} }
281                                     else { yyerror("Error : Expected an integer or float or double"); }
282                                 }}
283  ;
284  condx:  VARNAME EQUAL CONDX SEMI    { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable ");
yyerror($1); yyerror(" not declared\n"); exit(-1);}
285                       else { if(varmap[$1] == 5) { bvar[$1] = $3;}
286                             else { yyerror("Error : Expected bool");}
287                       }
288                     }
289      |TYPE VARNAME EQUAL CONDX SEMI  {
290                       if(strcmp($1,"bool") != 0) { yyerror("Error : A boolean expression must be placed in
a variable of type bool not "); yyerror($1); yyerror("\n"); exit(-1);}
291                       else bvar[$2] = $4;
292                     }
293  ;
294  CONDX:  NOT CONDX       { $$ = !($2) }
295      |CONDX AND A        { $$ = ($1 and $3.bval); }
296      |CONDX OR A     { $$ = ($1 or $3.bval); }
297      |A          { $$ = $1.bval; }
298  ;
299  A:  A EQUAL EQUAL B     { if($1.fval == $4) { $$.bval = true; } else { $$.bval = false; } }
300      |A NOT EQUAL B      { if($1.fval != $4) { $$.bval = true; } else { $$.bval = false; } }
301      |A GREATER B        { if($1.fval > $3) { $$.bval = true; } else { $$.bval = false; } }
302      |A GREATEREQ B      { if($1.fval >= $3) { $$.bval = true; } else { $$.bval = false; } }
303      |A LESS B       { if($1.fval < $3) { $$.bval = true; } else { $$.bval = false; } }
304      |A LESSEQ B     { if($1.fval <= $3) { $$.bval = true; } else { $$.bval = false; } }
305      |OPENBR1 CONDX CLOSEBR1 { $$.bval = $2; }
306      |B          { $$.fval = $1; }
307  ;
308  B:  OPENBR1 CONDX CLOSEBR1  { $$ = $2; }
309      |C          { $$ = $1; }
310  ;
311  C:  BOOLVAL         { $$ = $1;}
312      |DIGIT          { $$ = (float) ($1);}
313      |FLOATVAL       { $$ = $1;}
314      |VARNAME        { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable "); yyerror($1);
yyerror(" not declared\n");exit(-1);}
315                       else {
316                       if(varmap[$1] == 1) { $$ = (float) (ivar[$1]); }
317                       else if(varmap[$1] == 3) { currentex = 2; $$ = dvar[$1];}
318                       else if(varmap[$1] == 4) { currentex = 2; $$ = (float) (cvar[$1]);}
319                     }
320                   }
321      |CHARACTER      { $$ = (float)($1); }
322      |VARNAME OPENBR3 NUMVALUE CLOSEBR3  { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable
"); yyerror($1); yyerror(" not declared\n"); exit(-1); }
323                                 else {
324                                     if(varmap[$1] == 6) { if( $3 <= aivar[$1].size()){$$ =
(float)(aivar[$1][$3]);} else {yyerror("Error : Array index out of range"); exit(-1);} }
325                                     else if(varmap[$1] == 8) { if( $3 <= advar[$1].size()){$$ = advar[$1][$3];}
else {yyerror("Error : Array index out of range"); exit(-1);} }
326                                     else if(varmap[$1] == 9) { if( $3 <= acvar[$1].size()){$$ =
(float)(acvar[$1][$3]);} else {yyerror("Error : Array index out of range"); exit(-1);} }
327                                     else { yyerror("Error : Expected an integer or float or double"); }
328                                 }}
```

```
329  ;
330  IFELSE: IF CONDX COLON IFELSE else ENDIF    {  ifelfound = 0; if($2 == true) $$ = $4; else $$ = $5; }

331          |IF CONDX COLON IFELSE ENDIF { ifelfound = 0; if($2 == true) $$ = $4; else $$ = -100;       }
332          |EXP                { $$ = $1; }
333  ;
334  else:   ELIF CONDX COLON IFELSE else { if($2 == true){ $$ = $4; ifelfound = 1;}  else{$$ = $5;} }
335          |ELIF CONDX COLON IFELSE { if($2 == true){ $$ = $4; ifelfound = 1;} else{$$ = -100;}    }
336          |ELSE COLON IFELSE { if(ifelfound == 0) $$= $3; else $$ = -100; }
337  ;
338  FL: LOOP VARNAME IN NUMVALUE DOT DOT NUMVALUE BY NUMVALUE COLON EXP ENDLOOP { if(varmap.find($2) ==
varmap.end()) { yyerror("Error : Variable "); yyerror($2); yyerror(" not declared\n"); exit(-1); }
339                                                                 if(varmap[$2] != 1) { yyerror("Error : Expected
an integer in "); yyerror($2); yyerror("\n");}
340                                                                 cout << "//For loop detected\n";
341                                                                 if($4<=$7){
342                                                                 for(ivar[$2] = $4; ivar[$2] <= $7 ; ivar[$2] =
ivar[$2] + $9){
343                                                                   cout << "//Value of expression : " << $11 <<
endl;}
344                                                                 }
345                                                                 else{
346                                                                   for(ivar[$2] = $4; ivar[$2] >= $7 ; ivar[$2] =
ivar[$2] - $9){
347                                                                       cout << "//Value of expression : " << $11
<< endl;}
348                                                                 }
349                                                                 cout << endl;}
350  ;
351  NUMVALUE:   VARNAME { if(varmap.find($1) == varmap.end()) { yyerror("Error : Variable "); yyerror($1);
yyerror(" not declared\n");  exit(-1);}
352                       else { if(varmap[$1] != 1 ) { yyerror("Error : Expected an integer variable in ");
yyerror($1); yyerror("\n"); }
353                       else { $$ = ivar[$1]; }
354                     }
355                   }
356          |DIGIT          { $$ = $1; }
357          |DOUBLE     { yyerror("Error : Expected an integer"); }
358          |CHARACTER  { yyerror("Error : Expected an integer"); }
359          |BOOLVAL    { yyerror("Error : Expected an integer"); }
360          |VARNAME OPENBR3 NUMVALUE CLOSEBR3  { if(varmap.find($1) == varmap.end()) { yyerror("Error :
Variable "); yyerror($1); yyerror(" not declared\n");  exit(-1);}
361                              else {
362                                  if(varmap[$1] == 6) { if( $3 <= aivar[$1].size()){$$ = aivar[$1][$3];} else
{yyerror("Error : Array index out of range"); exit(-1);} }
363                                  else { yyerror("Error : Expected an integer"); }
364                              }}
365  ;
366  WH :        WHILE CONDX EXP ENDWHILE            { cout << "//while loop found." << endl;
367                                             //   while($2==true) {cout<< "//Value of expression :
"<<$3<<endl; }
368                                                   }
369  ;
370  singlecomment:  SCMNT               { cout << "//Single line comment" << endl; }
371  ;
372  multicomment:   MCMNT               { cout << "//Multiple line comment" << endl; }
373  ;
374  empty:  SEMI                    { cout<<"//empty statement"<<endl;}
375  ;
376  funcs :
377          |func funcs
378  ;
379  func :  FUNC TYPE VARNAME OPENBR1 PARAMLIST CLOSEBR1 exp returns FUNCEND  { cout << "//Function " << $3 <<
" declared." << endl; }
380  ;
381  exp :
```

```
        |EXP {cout<<"exp";}
;
PARAMLIST :
        |TYPE COLON VARNAME COMM PARAMLIST   { if(strcmp($1,"int") == 0) { varmap[$3] = 1; ivar[$3] = 0; }
                                        else if(strcmp($1,"double") == 0) {varmap[$3] = 3; dvar[$3] = 0.0;
}
                                        else if(strcmp($1,"char") == 0) {varmap[$3] = 4; cvar[$3] = '#'; }
                                        else if(strcmp($1,"bool") == 0) {varmap[$3] = 5; bvar[$3] = true; }
                                        }
        |TYPE COLON VARNAME {  if(strcmp($1,"int") == 0) {varmap[$3] = 1; ivar[$3] = 0; }
                        else if(strcmp($1,"double") == 0) {varmap[$3] = 3; dvar[$3] = 0.0; }
                        else if(strcmp($1,"char") == 0) {varmap[$3] = 4; cvar[$3] = '#'; }
                        else if(strcmp($1,"bool") == 0) {varmap[$3] = 5; bvar[$3] = true; }
                        cout<<varmap[$3]<<endl;
                        }
;
returns:    RETURN VALUE SEMI
        |RETURN VARNAME SEMI            { if(varmap[$2]==0){ //if( varmap.find($2) == varmap.end()) {
            yyerror("Error : Variable "); yyerror($2); yyerror(" not declared\n"); }}
;
fcall : GCD OPENBR1 NUMVALUE COMM NUMVALUE CLOSEBR1 SEMI { cout<<"GCD : "<< __gcd($3,$5) <<endl; }
        | LCM OPENBR1 NUMVALUE COMM NUMVALUE CLOSEBR1 SEMI { cout<<"LCM :
"<<(($3)*(($5)/__gcd($3,$5)))<<endl;}
        | SQRT OPENBR1 FLOATVAL CLOSEBR1 SEMI { cout<<"Square root of value : "<<sqrt($3)<<endl; }
        | SQ OPENBR1 FLOATVAL CLOSEBR1 SEMI {cout<<"Square of value :"<<$3*$3<<endl;}
        | QUBE OPENBR1 FLOATVAL CLOSEBR1 SEMI  {cout<<"Qube of value :"<<$3*$3*$3<<endl;}
        | ISPRM OPENBR1 NUMVALUE CLOSEBR1 SEMI {siv(); if(($3 % 2 == 1 && isprime[$3]==0 ) || $3 ==
2)cout<<$3<<" is prime."<<endl;
                                                 else cout<<$3<<" is not a prime."<<endl; }
        | PRIME OPENBR1 NUMVALUE CLOSEBR1 SEMI {siv(); cout<<$3<<" th prime number is : "<<pr[$3 - 1
]<<endl;}
        | FIBO OPENBR1 NUMVALUE CLOSEBR1 SEMI { if($3<3) cout<<"1st Fibonacci number is 1"<<endl;
                                else{ int x=1,y=1,z;for(int i=0;i< $3 - 2 ;i++){ z=x+y; x=y;y=z; }
cout<<$3<<" th Fibonacci number is : "<<z<<endl;} }
        | POW OPENBR1 NUMV COMM NUMV CLOSEBR1 SEMI { cout<<$3<<" to the power "<<$5<<" is :
"<<powl($3,$5)<<endl; }
        | BIGM OPENBR1 NUMVALUE COMM NUMVALUE COMM NUMVALUE CLOSEBR1 SEMI {cout<<"Bigmod value :
"<<bigmod($3,$5,$7)<<endl;}
        | MOD OPENBR1 NUMVALUE COMM NUMVALUE CLOSEBR1 SEMI { cout<<"Modulus Result : "<<$3 % $5<<endl;}
        | SIN OPENBR1 FLOATVAL CLOSEBR1 SEMI { double x=pi/180.0; cout<<"Sin"<<$3<<" is : "<<sin($3 *
x)<<endl; }
        | COS OPENBR1 FLOATVAL CLOSEBR1 SEMI { double x=pi/180.0; cout<<"Cos"<<$3<<" is : "<<cos($3)<<endl;
}
        | TAN OPENBR1 FLOATVAL CLOSEBR1 SEMI { double x=pi/180.0; cout<<"Tan"<<$3<<" is : "<<tan($3 *
x)<<endl; }
        | ASIN OPENBR1 FLOATVAL CLOSEBR1 SEMI { cout<<"Sin invers "<<$3<<" is : "<<asin($3)/pi*180.0<<endl;
}
        | ACOS OPENBR1 FLOATVAL CLOSEBR1 SEMI { cout<<"Cos invers "<<$3<<" is : "<<acos($3)/pi*180.0<<endl;
}
        | ATAN OPENBR1 FLOATVAL CLOSEBR1 SEMI { cout<<"Tan invers "<<$3<<" is : "<<atan($3)/pi*180.0<<endl;
}
;
NUMV : DIGIT    {$$ = $1;}
       | FLOATVAL {$$ = $1;}
;
%%

int main() {
  yyin = fopen("cmpin.txt","r");
  yyout = freopen("cmpout.txt","w",stdout);
  yyparse();
}
void yyerror(const char *s) {
  cout << s ;
}

```