# Syntax Test Script

## General Structure of the Program

Syntax: program = <start><function><main><function><end>;;

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
PrimaryMission() {	No Error	No Error	OK
<pre>} deploy();</pre>			
PrimaryMission() {	No Error	No Error	OK
}			
<pre>unit a() { } PrimaryMission() {</pre>	unexpected token "}", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(",	<pre>unexpected token "}", expected one of "post", "captured", "inquire", "inorder", "go", "phase",</pre>	OK
} deploy();	"Numlit", "Declit", "id", "Stringlit", "joe", "++", or "", on line: 2 column: 1	"campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "", on line: 2 column: 1	
<pre>unit a() { post("Function1"); } PrimaryMission() {</pre>	unexpected token "}", expected "backup", on line: 3 column: 1	unexpected token "}", expected "backup", on line: 3 column: 1	OK
<pre>} deploy();</pre>			
<pre>unit a() { post("Function1"); backup(); }</pre>	unexpected token ")", expected "id", on line: 3 column: 8	unexpected token ")", expected "id", on line: 3 column: 8	OK
PrimaryMission() {			
<pre>} deploy();</pre>			
<pre>unit a() { post("Function1");</pre>	No Error	No Error	OK

<pre>backup(a); } PrimaryMission() { } deploy();</pre>			
deploy();	unexpected token "deploy", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	unexpected token "deploy", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	OK
hold unit a=0;	No Error	No Error	OK
PrimaryMission() {			
} deploy();			
PrimaryMission() {	unexpected token "PrimaryMission",	unexpected token "PrimaryMission",	OK
}	expected "deploy", on	expected "deploy", on	
PrimaryMission() {	line: 3 column: 1	line: 3 column: 1	
<pre>} deploy();</pre>			
Mission() {	unexpected token "id",	unexpected token "id",	OK
unit a=0;	expected one of "hold", "unit", "digit",	expected one of "hold", "unit",	
,	"company", "joe",	"digit", "company",	
PrimaryMission() {	"response", "miss",	"joe", "response",	
} deploy();	"struct", or "PrimaryMission", on	"miss", "struct", or "PrimaryMission", on	
	line: 2 column: 1	line: 2 column: 1	
miss Mission() {	unexpected token	unexpected token	OK
unit a=0;	"backup", expected one of "post", "captured",	"backup", expected one	
backup(a); }	post, captured, "inquire", "inorder",	of "post", "captured", "inquire", "inorder",	
J	"go", "phase",	"go", "phase",	
<pre>PrimaryMission() {</pre>	"campaign", "(",	"campaign", "(",	
	"Numlit", "Declit", "id",	"Numlit", "Declit",	

} deploy();	"Stringlit", "joe", "++", or "", on line: 3 column: 1	"id", "Stringlit", "joe", "++", or "", on line: 3 column: 1	
<pre>miss Mission() { unit a=0; post("Mission"); }  PrimaryMission() { } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission(unit a) { } deploy();</pre>	unexpected token "unit", expected ")", on line: 1 column: 16	unexpected token "unit", expected ")", on line: 1 column: 16	OK
<pre>unit a(unit c) { post("function1"); backup(c); } unit d(unit e) { post("function2"); backup(e); } PrimaryMission() { } deploy();</pre>	No Error	No Error	OK
<pre>unit a(unit c) { post("function1"); backup(c); } miss d(unit e) { post("function2"); } PrimaryMission() { } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { unit a=0; company="String"; } deploy();</pre>	No Error	No Error	OK

<pre>PrimaryMission() { unit a=0; "String"; } deploy();</pre>	unexpected token ";", expected "}", on line: 3 column: 10	<pre>unexpected token ";", expected "}", on line: 3 column: 10</pre>	OK
<pre>PrimaryMission() { unit a() { post("function"); backup(id); } deploy();</pre>	No Error	No Error	OK
<pre>post("Hello"); PrimaryMission() { } deploy();</pre>	unexpected token "post", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	unexpected token "post", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	OK

## Constant Declaration

Syntax: <const> = hold <datatype> <identifier> := <initValue>;

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
hold unit a=0;	No Error	No Error	OK
hold unit a;	unexpected token ";",	unexpected token ";",	OK
	expected "=", on line: 1	expected "=", on line:	
	column: 11	1 column: 11	
hold a=0;	unexpected token "id",	unexpected token "id",	OK
	expected one of "unit",	expected one of	
	"digit", "joe",	"unit", "digit",	
	"company", "response", or	"joe", "company",	
	";", on line: 1 column: 5	"response", or ";", on	
		line: 1 column: 5	
hold unit a="Hello";	unexpected token	unexpected token	OK
	"Stringlit", expected	"Stringlit", expected	

	"Numlit", on line: 1	"Numlit", on line: 1	
	column: 12	column: 12	
hold digit a=1	unexpected token	unexpected token	OK
	"Numlit", expected	"Numlit", expected	
	"Declit", on line: 1	"Declit", on line: 1	
	column: 13	column: 13	
hold digit a=1.55, c=1.767;			OK
	unexpected token ",",	unexpected token ",",	
	expected ";", on line: 1	expected ";", on line:	
	column: 19	1 column: 19	

## Global and Local variable declaration

Syntax: <varDec> = <datatype> <identifier> <inittype>;

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
unit a=0;	No Error	No Error	OK
company Welcome="Hello";	No Error	No Error	OK
digit a=1.555;	No Error	No Error	OK
joe char='\n';	No Error	No Error	OK
response isTaken = NEGATIVE;	No Error	No Error	OK
unit a, b = 2, c;	No Error	No Error	OK
<pre>company Message, Name = "Ron", Set;</pre>	No Error	No Error	OK
unit Num = 1.008;	unexpected token "Declit", expected	unexpected token "Declit", expected	OK

	"Numlit", on line: 1	"Numlit", on line: 1	
	column: 8	column: 8	
joe Message = "HelloWorld";	unexpected token	unexpected token	OK
	"Stringlit", expected	"Stringlit", expected	
	"Charlit", on line: 1	"Charlit", on line: 1	
	column: 7	column: 7	
joe Message = 1.000;	unexpected token	unexpected token	OK
	"Declit", expected	"Declit", expected	
	"Charlit", on line: 1	"Charlit", on line: 1	
	column: 7	column: 7	
unit id = ;	unexpected token ";",	unexpected token ";",	OK
	expected "Numlit", on	expected "Numlit", on	
	line: 1 column: 8	line: 1 column: 8	
unit Emnum = ,2;	unexpected token ",",	unexpected token ",",	OK
	expected "Numlit", on	expected "Numlit", on	
	line: 1 column: 8	line: 1 column: 8	

# Array Declaration

Syntax: <varDec> = <datatype> <identifier> <N1> <N2> <varElem>;

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
unit id[5][6] = {1, 3};	No Error	No Error	OK
unit a[1] = {2};	No Error	No Error	OK
unit a[6+2] = {2, 4,6,7,9, 1, 3, 4};	No Error	No Error	OK
digit Decimals[4] = {1.5, 2.6, 1.2, 4.6, 0.23};	No Error	No Error	OK
<pre>digit Decimals[4] = {1, 3, 5, 7, 4 };</pre>	unexpected token "Numlit", expected "}", on line: 1 column: 18	unexpected token "Numlit", expected "}", on line: 1 column: 18	OK
<pre>company phrase[4] = { "cat",     "sat", "on", "mat" };</pre>	No Error	No Error	OK

<pre>company phrase[4] = { "cat",     "sat", "on", 'm' };</pre>	unexpected token "Charlit", expected "Stringlit", on line: 1 column: 50	unexpected token "Charlit", expected "Stringlit", on line: 1 column: 50	OK
joe chars[2] = { 'a', 'b' };	No Error	No Error	OK
joe chars[2] = { 'a', "HI" };	<pre>unexpected token "Stringlit", expected "}", on line: 1 column: 24</pre>	<pre>unexpected token "Stringlit", expected "}", on line: 1 column: 24</pre>	OK
<pre>PrimaryMission() { unit a[3][5] = 32; } deploy();</pre>	unexpected token "Numlit", expected "{", on line: 2 column: 24	<pre>unexpected token "Numlit", expected "{", on line: 2 column: 24</pre>	OK
PrimaryMission() { unit a[3][5][6] = {4.6,7}; } deploy();	unexpected token "[", expected ";", on line: 2 column: 23	unexpected token "[", expected ";", on line: 2 column: 23	OK

## Post and Captured Declaration

Syntax:

```
<print> = post ( id || Stringlit ) ; <print>
<scan> = capture (#id);
```

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>post("HelloWorld!");</pre>	No Error	No Error	OK
post(Name);	No Error	No Error	OK
post(1);	unexpected token	unexpected token	OK
	"Numlit", expected one of	"Numlit", expected one	
	"Stringlit", or "id", on	of "Stringlit", or	
	line: 2 column: 6		

		"id", on line: 2	
		column: 6	
<pre>post("One"); post("Two"); post("Three");</pre>	No Error	No Error	OK
post("Hello")	<pre>unexpected token "}", expected ";", on line: 3 column: 1</pre>	<pre>unexpected token "}", expected ";", on line: 3 column: 1</pre>	OK
<pre>captured(#num);</pre>	No Error	No Error	OK
captured(1num);	unexpected token "captured", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	unexpected token "captured", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	OK
<pre>PrimaryMission() {   captured(#num) } deploy();</pre>	<pre>unexpected token "}", expected ";", on line: 3 column: 1</pre>	<pre>unexpected token "}", expected ";", on line: 3 column: 1</pre>	OK
<pre>PrimaryMission() { captured(1num); } deploy();</pre>	unexpected token "Numlit", expected "#", on line: 2 column: 10	unexpected token "Numlit", expected "#", on line: 2 column: 10	OK
<pre>captured(#num); PrimaryMission() { } deploy();</pre>	unexpected token "captured", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	unexpected token "captured", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	OK
<pre>PrimaryMission() { } deploy(); captured(#num);</pre>	unexpected token "captured", expected <eof>, on line: 3 column:</eof>	unexpected token "captured", expected <eof>, on line: 3 column: 1</eof>	OK
PrimaryMission() {	No Error	No Error	OK

```
captured (#num);
captured (#name);
captured (#addr);
deploy();
```

#### Function Declaration

```
Syntax:
<funct> = <dtype> ( dtypeA ) { <statement> } ; <body>
```

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>unit MissionOne(unit a, unit b, unit c) { post("Mission1"); backup(a); }</pre>	No Error	No Error	OK
<pre>unit MissionOne(unit a, unit b, unit c) { post("Mission1"); backup(a); } unit Missiontwo(unit a) { post("Mission2"); backup(a); }</pre>	No Error	No Error	OK
<pre>miss MissionOne(unit a, unit b, unit c) { post("Mission1"); backup(a); }</pre>	unexpected token "backup", expected "}", on line: 3 column: 1	<pre>unexpected token "backup", expected "}", on line: 3 column: 1</pre>	OK

```
unit Missiontwo(unit a) {
post("Mission2");
backup(a);
                                                                                      OK
miss MissionOne(unit a, unit
                                 No Error
                                                             No Error
b, unit c) {
post("MissionMiss");
miss MissionOne (unit a, unit
                                 No Error
                                                             No Error
                                                                                      OK
b, unit c) {
post("MissionMiss");
captured (#num);
                                                                                      OK
miss MissionOne (unit a, unit
                                 No Error
                                                             No Error
b, unit c) {
unit a=0;
joe a = 'd';
company="abortmission";
miss MissionOne (unit a, unit
                                 unexpected token
                                                             unexpected token
                                                                                      OK
                                                             "captured", expected
) {
                                 "captured", expected one
                                 of "hold", "unit",
                                                             one of "hold", "unit",
unit a=0;
                                 "digit", "company",
                                                             "digit", "company",
joe a = 'd';
                                 "joe", "response",
                                                             "joe", "response",
company="abortmission";
                                 "miss", "struct", or
                                                             "miss", "struct", or
                                                             "PrimaryMission", on
                                 "PrimaryMission", on
                                 line: 1 column: 1
                                                             line: 1 column: 1
                                 No Error
miss MissionOne(unit a[] ) {
                                                             No Error
                                                                                      OK
unit a=0;
joe a = 'd';
company="abortmission";
```

```
miss MissionOne(unit a[],
                                No Error
                                                           No Error
                                                                                    OK
unit c[] ) {
unit a=0;
joe a = 'd';
company="abortmission";
unit b=0;
                                No Error
                                                           No Error
                                                                                    OK
miss MissionOne(unit a[] ,
unit c[] ) {
unit a=0;
joe a = 'd';
company="abortmission";
                                unexpected token "=",
                                                           unexpected token "=",
miss MissionOne(unit a[] ,
                                                                                    OK
                                expected "(", on line: 6
                                                           expected "(", on line:
unit c[] ) {
                                column: 7
                                                           6 column: 7
unit a=0;
joe a = 'd';
company="abortmission";
unit b=0;
```

## Assignments

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>PrimaryMission() { num=num+num; } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { num=num+num*num/num-num; } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { num=num^4; } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { num = 1; } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() {   i++; } deploy();</pre>	No Error	No Error	OK

<pre>PrimaryMission() { ++i; } deploy();</pre>	No Error	No Error	OK
PrimaryMission() { a = (a+3-45) + (2-5+6/2); } deploy();	No Error	No Error	OK
PrimaryMission() { a=(a-2+5/8*b) -(153^a); } deploy();	No Error	No Error	OK
<pre>PrimaryMission() { +a; } deploy();</pre>	<pre>unexpected token "+", expected "}", on line: 2 column: 1</pre>	<pre>unexpected token "+", expected "}", on line: 2 column: 1</pre>	OK
PrimaryMission() { a+=3+56; } deploy();	No Error	No Error	OK
PrimaryMission() { a=(a-2+5>8!b) -(153^a); } deploy();	unexpected token ">", expected ")", on line: 2 column: 21	unexpected token ">", expected ")", on line: 2 column: 21	OK
<pre>PrimaryMission() { num=num=num=num=num; } deploy();</pre>	<pre>unexpected token ";", expected "}", on line: 2 column: 18</pre>	unexpected token ";", expected "}", on line: 2 column: 18	OK

# Inquire statement

```
Syntax:  \langle \text{for} \rangle = \text{inquire (id = Numlit ; id } \langle \text{op1} \rangle \text{ Numlit ; } \langle \text{mntcond} \rangle \text{ ; ) } \{ \langle \text{statement} \rangle \}
```

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>PrimaryMission() { inquire(i=0;i&lt;= 1;i++;) { } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { inquire(i=m;i&lt;= 1;i++; ) { } deploy();</pre>	unexpected token "id", expected one of "Numlit", or "Zero", on line: 2 column: 12 unexpected token "++", expected one of "=", "<", ">", "!", "<=", or ">=", on line: 2 column: 28	unexpected token "id", expected one of "Numlit", or "Zero", on line: 2 column: 12 unexpected token "++", expected one of "=", "<", ">", "!", "<=", or ">=", on line: 2 column: 28	OK
<pre>PrimaryMission() { inquire(i=0;i&lt;=1; ++i; ) { } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { inquire(i=0;i&lt;=1; ++i; ) { i++; } deploy();</pre>	No Error	No Error	OK

```
PrimaryMission() {
                                 unexpected token "id",
                                                              unexpected token "id",
                                                                                        OK
inquire(i=0; i \le 1; ++i; ) i++;
                                 expected "{", on line: 2
                                                              expected "{", on line:
                                 column: 36
                                                              2 column: 36
} deploy();
PrimaryMission() {
                                                                                        0K
                                 No Error
                                                              No Error
inquire(i=0; i \le 1; ++i;) {
num=num+1;
} deploy();
                                 unexpected token "{",
                                                              unexpected token "{",
PrimaryMission() {
                                                                                        OK
                                 expected "}", on line: 2
                                                              expected "}", on line:
inquire(i=0; i <=1; ++i;) { {
                                                              2 column: 37
num=num+1;
                                 column: 37
} deploy();
PrimaryMission() {
                                 No Error
                                                              No Error
                                                                                        OK
inquire(i=0; i \le 1; ++m;) {
num=num+1;
} deploy();
                                 unexpected token "+",
                                                              unexpected token "+",
PrimaryMission() {
                                                                                        OK
                                 expected one of "++", "--
                                                              expected one of "++",
inquire(i=0; i \le 1; +i;) {
                                  ", or "id", on line: 2
                                                              "--", or "id", on
                                 column: 30
                                                              line: 2 column: 30
} deploy();
                                 unexpected token "=",
PrimaryMission() {
                                                              unexpected token "=",
                                                                                        OK
inquire(i==0; i \le 1; ++i;) {
                                 expected one of "Numlit",
                                                              expected one of
                                 or "Zero", on line: 2
                                                              "Numlit", or "Zero",
                                 column: 12
                                                              on line: 2 column: 12
} deploy();
inquire(i=0; i \le 1; ++i;) {
                                 unexpected token
                                                              unexpected token
                                                                                        OK
                                 "inquire", expected one
                                                              "inquire", expected
                                 of "hold", "unit",
                                                              one of "hold", "unit",
PrimaryMission() {
                                                              "digit", "company",
                                 "digit", "company",
} deploy();
```

	"joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	"joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	
<pre>PrimaryMission() { } deploy(); inquire(i=0;i&lt;=1; ++i; ) { }</pre>	unexpected token "inquire", expected <eof>, on line: 3 column: 1</eof>	unexpected token "inquire", expected <eof>, on line: 3 column: 1</eof>	OK

#### Inorder, otherorder and order statement

```
Syntax:
```

```
<ifelse> = inorder(<Relop> || <Logop>) { <statement> } <elseif> <else> <ifelse>
<elseif> = otherorder(Relop || Logop) { <statement> } <elseif> <else> <ifelse>
<else> = order { <statement> } <ifelse>
```

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>PrimaryMission() { inorder(i &lt;= 0) { } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { inorder(i &lt;= 0    i &gt; 0) { } deploy();</pre>	unexpected token "  ", expected ")", on line: 2 column: 19	unexpected token "  ", expected ")", on line: 2 column: 19	OK
PrimaryMission() { inorder((i <= 0)    (i > 0)) {	No Error	No Error	OK

```
} deploy();
PrimaryMission() {
                                  unexpected token "{",
                                                              unexpected token "{",
                                                                                        OK
inorder((i \le 0) \mid | (i > 0)  {
                                  expected ")", on line: 2
                                                              expected ")", on line:
                                  column: 34
                                                              2 column: 34
} deploy();
PrimaryMission() {
                                  No Error
                                                              No Error
                                                                                         0K
inorder((i \le 0) & (i > 0))  {
} deploy();
                                  unexpected token ")",
                                                              unexpected token ")",
PrimaryMission() {
                                                                                         OK
inorder((i \le 0) || (i = 0) &
                                  expected "(", on line: 2
                                                              expected "(", on line:
(b>5) & ) {
                                                              2 column: 48
                                  column: 48
} deploy();
PrimaryMission() {
                                  No Error
                                                              No Error
                                                                                         OK
inorder((i \le 0) \mid | (i = 0) \&
(b>5) & (m!0)
} deploy();
PrimaryMission() {
                                                              unexpected token "(",
                                  unexpected token "(",
                                                                                        OK
inorder((((((((i<=0) | (i==0))
                                  expected one of "Numlit",
                                                              expected one of
& (b>5) & (m!0)) {
                                  "Declit", "id",
                                                               "Numlit", "Declit",
                                  "Stringlit", or "joe", on
                                                              "id", "Stringlit", or
                                                               "joe", on line: 2
} deploy();
                                  line: 2 column: 10
                                                              column: 10
PrimaryMission()
                                  No Error
                                                              No Error
                                                                                        OK
```

```
inorder((i <=0) || (i ==0) &
(b>5) & (m!0) 
post("ifelse");
} deploy();
inorder((i \le 0) \mid | (i = 0) \&
                                 unexpected token
                                                              unexpected token
                                                                                        OK
(b>5) & (m!0)
                                  "inorder", expected one
                                                              "inorder", expected
                                 of "hold", "unit",
                                                              one of "hold", "unit",
post("ifelse");
                                 "digit", "company",
                                                              "digit", "company",
                                  "joe", "response",
PrimaryMission() {
                                                              "joe", "response",
                                                              "miss", "struct", or
} deploy();
                                  "miss", "struct", or
                                  "PrimaryMission", on
                                                              "PrimaryMission", on
                                  line: 1 column: 1
                                                              line: 1 column: 1
PrimaryMission() {
                                  unexpected token
                                                                                        OK
                                                              unexpected token
                                  "inorder", expected
} deploy();
                                                              "inorder", expected
inorder((i <=0) || (i ==0) &
                                  <EOF>, on line: 3 column:
                                                              <EOF>, on line: 3
(b>5) & (m!0)
                                                              column: 1
post("ifelse");
PrimaryMission() {
                                 No Error
                                                              No Error
                                                                                        OK
inorder (i \le 0) {
post("ifelse");
otherorder(i \ge 5) {
post("elseif");
} deploy();
PrimaryMission() {
                                 No Error
                                                              No Error
                                                                                        OK
inorder (i \le 0) {
post("ifelse");
otherorder(i \ge 5) {
post("elseif");
otherorder(j!=10) {
post("elseif2");
} deploy();
```

<pre>PrimaryMission() {   otherorder(j!=10) {   post("elseif2");   }   inorder(i&lt;=0) {   post("ifelse");   }   otherorder(i&gt;=5) {   post("elseif");   } } deploy();</pre>	<pre>unexpected token "otherorder", expected "}", on line: 2 column: 1</pre>	unexpected token "otherorder", expected "}", on line: 2 column: 1	OK
<pre>PrimaryMission() {   inorder(i &lt;= 0) {   post("ifelse");   }   otherorder(i &gt;= 5) {   post("elseif");   }   otherorder(j!=m    ) {   post("elseif2");   }   } deploy();</pre>	unexpected token "  ", expected ")", on line: 8 column: 18	unexpected token "  ", expected ")", on line: 8 column: 18	OK
<pre>PrimaryMission() { inorder(i &lt;= 0) { post("ifelse"); } otherorder(i &gt;= 5) { post("elseif"); } otherorder((j!=m)    (1 &lt; 0)) { post("elseif2"); } } deploy();</pre>	No Error	No Error	OK
PrimaryMission() {	No Error	No Error	OK

```
inorder (i \le 0) {
post("ifelse");
otherorder(i>=5) {
post("elseif");
otherorder ((j!=m) \mid | (1<0)) {
post("elseif2");
order {
post("else");
} deploy();
                                 unexpected token "(",
                                                              unexpected token "(",
PrimaryMission() {
                                                                                        OK
                                  expected "{", on line: 11
                                                              expected "{", on line:
inorder (i \le 0) {
                                                               11 column: 6
                                  column: 6
post("ifelse");
otherorder(i \ge 5) {
post("elseif");
otherorder((j!=m) || (1<0)) {
post("elseif2");
order (i=3) {
post("else");
} deploy();
PrimaryMission() {
                                  unexpected token "order",
                                                              unexpected token
                                                                                        OK
                                  expected <EOF>, on line:
                                                               "order", expected
inorder (i \le 0) {
                                  12 column: 1
                                                               <EOF>, on line: 12
post("ifelse");
                                                               column: 1
otherorder(i \ge 5) {
post("elseif");
otherorder((j!=m) | (1<0)) {
post("elseif2");
} deploy();
order {
post("else");
```

```
PrimaryMission() {
                                  unexpected token "order",
                                                               unexpected token
                                                                                         OK
                                  expected "}", on line: 2
                                                               "order", expected "}",
order {
post("else");
                                                               on line: 2 column: 1
                                  column: 1
inorder (i \le 0) {
post("ifelse");
otherorder(i \ge 5) {
post("elseif");
otherorder ((j!=m) \mid | (1<0)) {
post("elseif2");
} deploy();
PrimaryMission() {
                                  unexpected token "post",
                                                               unexpected token
                                                                                         OK
                                  expected "{", on line: 12
                                                               "post", expected "{",
inorder (i \le 0) {
                                  column: 1
                                                               on line: 12 column: 1
post("ifelse");
otherorder (i \ge 5) {
post("elseif");
otherorder ((j!=m) \mid | (1<0)) {
post("elseif2");
order
post("else");
} deploy();
```

#### go phase and phase statement

```
Syntax:

<dowhile> = go { <statement> } phase( <RelOp> ) ; <body>

<while> = phase( <RelOp> ) { <statement> } <body>
```

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>PrimaryMission() { go { } phase(a!=0); } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { go { unit a=0; response isGo=AFFIRMATIVE; post("go"); } phase(a!=0); } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { go { unit a=0; response isGo=AFFIRMATIVE; } phase(a!=0); } deploy();</pre>	unexpected token "}", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "", on line: 5 column: 1 unexpected token ";", expected "}", on line: 5 column: 19	unexpected token "}", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "", on line: 5 column: 1 unexpected token ";", expected "}", on line: 5 column: 19	OK
<pre>PrimaryMission() { go { post("GoFirst"); go { post("GoSecond"); } phase(b&gt;=0); } phase(a!=0); } deploy();</pre>	No Error	No Error	OK

<pre>PrimaryMission() { go { post("GoFirst"); go { post("GoSecond"); } phase(a!=0); } deploy(); } phase(b&gt;=0);</pre>	unexpected token "deploy", expected "phase", on line: 7 column: 2	unexpected token "deploy", expected "phase", on line: 7 column: 2	OK
<pre>PrimaryMission() { go { post("GoFirst"); } phase(b&gt;=0); go { post("GoSecond"); } phase(a!=0); } deploy();</pre>	No Error	No Error	OK
<pre>} phase(b&gt;=0); PrimaryMission() { go { post("GoFirst"); go { post("GoSecond"); } phase(a!=0); } deploy();</pre>	unexpected token "}", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	unexpected token "}", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1	OK
<pre>PrimaryMission() { phase(i&lt;90) { post("Average"); } deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() { phase(i&lt;90) { post("Average"); phase(i&gt;90) { post("Low"); } } deploy();</pre>	No Error	No Error	OK

# Campaign and operation statement

```
Syntax:
<switch> = campaign( id ) { <statement> } <body>
<case> = operation id : <statement> abort();
```

SAMPLE INPUT	SYSTEM Expected Output	SYSTEM Actual Output	Remarks
<pre>PrimaryMission() {   campaign(num) {    operation a:   post("operation1");   abort();   operation b:   post("operation2");   abort(); }   deploy();</pre>	No Error	No Error	OK
<pre>PrimaryMission() {   campaign(num) {    operation a:   post("operation1");   abort();   operation b:   post("operation2"); }   deploy();</pre>	unexpected token "}", expected "abort", on line: 8 column: 1	unexpected token "}", expected "abort", on line: 8 column: 1	OK
<pre>PrimaryMission() {   operation a:   post("operation1");   abort();</pre>	<pre>unexpected token "operation", expected "}", on line: 2 column: 1</pre>	<pre>unexpected token "operation", expected "}", on line: 2 column: 1</pre>	OK

```
campaign(num) {
operation b:
post("operation2");
abort();
} deploy();
PrimaryMission() {
                                                                                      OK
                                 unexpected token
                                                             unexpected token
                                 "campaign", expected
                                                             "campaign", expected
} deploy();
                                 <EOF>, on line: 3 column:
                                                             <EOF>, on line: 3
                                                             column: 1
campaign (num) {
                                 1
operation a:
post("operation1");
abort();
operation b:
post("operation2");
abort();
PrimaryMission() {
                                 No Error
                                                             No Error
                                                                                      OK
campaign(num) {
operation a:
post("operation1");
campaign(num2) {
operation ab:
post("operation1.5");
abort();
abort();
operation b:
post("operation2");
abort();
} deploy();
```