**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() {  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  } | No Error | No Error | **OK** |
| unit a() {  }  PrimaryMission() {  } deploy(); | unexpected token "}", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "--", on line: 2 column: 1 | unexpected token "}", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "--", on line: 2 column: 1 | **OK** |
| unit a() {  post("Function1");  }  PrimaryMission() {  } deploy(); | unexpected token "}", expected "backup", on line: 3 column: 1 | unexpected token "}", expected "backup", on line: 3 column: 1 | **OK** |
| unit a() {  post("Function1");  backup();  }  PrimaryMission() {  } deploy(); | unexpected token ")", expected "id", on line: 3 column: 8 | unexpected token ")", expected "id", on line: 3 column: 8 | **OK** |
| unit a() {  post("Function1");  backup(a);  }  PrimaryMission() {  } deploy(); | No Error | No Error | **OK** |
| 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;  PrimaryMission() {  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  }  PrimaryMission() {  } deploy(); | unexpected token "PrimaryMission", expected "deploy", on line: 3 column: 1 | unexpected token "PrimaryMission", expected "deploy", on line: 3 column: 1 | **OK** |
| Mission() {  unit a=0;  }  PrimaryMission() {  } deploy(); | unexpected token "id", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 2 column: 1 | unexpected token "id", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 2 column: 1 | **OK** |
| miss Mission() {  unit a=0;  backup(a);  }  PrimaryMission() {  } deploy(); | unexpected token "backup", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "--", on line: 3 column: 1 | unexpected token "backup", expected one of "post", "captured", "inquire", "inorder", "go", "phase", "campaign", "(", "Numlit", "Declit", "id", "Stringlit", "joe", "++", or "--", on line: 3 column: 1 | **OK** |
| miss Mission() {  unit a=0;  post("Mission");  }  PrimaryMission() {  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission(unit a) {  } deploy(); | unexpected token "unit", expected ")", on line: 1 column: 16 | unexpected token "unit", expected ")", on line: 1 column: 16 | **OK** |
| unit a(unit c) {  post("function1");  backup(c);  }  unit d(unit e) {  post("function2");  backup(e);  }  PrimaryMission() {  } deploy(); | No Error | No Error | **OK** |
| unit a(unit c) {  post("function1");  backup(c);  }  miss d(unit e) {  post("function2");  }  PrimaryMission() {  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  unit a=0;  company="String";  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  unit a=0;  "String";  } deploy(); | unexpected token ";", expected "}", on line: 3 column: 10 | unexpected token ";", expected "}", on line: 3 column: 10 | **OK** |
| PrimaryMission() {  unit a() {  post("function");  backup(id);  }  } deploy(); | No Error | No Error | **OK** |
| post("Hello");  PrimaryMission() {  } deploy(); | 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 ";", expected "=", on line: 1 column: 11 | unexpected token ";", expected "=", on line: 1 column: 11 | **OK** |
| hold a=0; | unexpected token "id", expected one of "unit", "digit", "joe", "company", "response", or ";", on line: 1 column: 5 | unexpected token "id", expected one of "unit", "digit", "joe", "company", "response", or ";", on line: 1 column: 5 | **OK** |
| hold unit a="Hello"; | unexpected token "Stringlit", expected "Numlit", on line: 1 column: 12 | unexpected token "Stringlit", expected "Numlit", on line: 1 column: 12 | **OK** |
| hold digit a=1 | unexpected token "Numlit", expected "Declit", on line: 1 column: 13 | unexpected token "Numlit", expected "Declit", on line: 1 column: 13 | **OK** |
| hold digit a=1.55, c=1.767; | unexpected token ",", expected ";", on line: 1 column: 19 | unexpected token ",", expected ";", on line: 1 column: 19 | **OK** |

**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** |
| company Message, Name = "Ron", Set; | No Error | No Error | **OK** |
| unit Num = 1.008; | unexpected token "Declit", expected "Numlit", on line: 1 column: 8 | unexpected token "Declit", expected "Numlit", on line: 1 column: 8 | **OK** |
| joe Message = "HelloWorld"; | unexpected token "Stringlit", expected "Charlit", on line: 1 column: 7 | unexpected token "Stringlit", expected "Charlit", on line: 1 column: 7 | **OK** |
| joe Message = 1.000; | unexpected token "Declit", expected "Charlit", on line: 1 column: 7 | unexpected token "Declit", expected "Charlit", on line: 1 column: 7 | **OK** |
| unit id = ; | unexpected token ";", expected "Numlit", on line: 1 column: 8 | unexpected token ";", expected "Numlit", on line: 1 column: 8 | **OK** |
| unit Emnum = ,2; | unexpected token ",", expected "Numlit", on line: 1 column: 8 | unexpected token ",", expected "Numlit", on line: 1 column: 8 | **OK** |

**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** |
| digit Decimals[4] = {1, 3, 5, 7, 4 }; | unexpected token "Numlit", expected "}", on line: 1 column: 18 | unexpected token "Numlit", expected "}", on line: 1 column: 18 | **OK** |
| company phrase[4] = { "cat", "sat", "on", "mat" }; | No Error | No Error | **OK** |
| company phrase[4] = { "cat", "sat", "on", 'm' }; | 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" }; | unexpected token "Stringlit", expected "}", on line: 1 column: 24 | unexpected token "Stringlit", expected "}", on line: 1 column: 24 | **OK** |
| PrimaryMission() {  unit a[3][5] = 32;  } deploy(); | unexpected token "Numlit", expected "{", on line: 2 column: 24 | unexpected token "Numlit", expected "{", on line: 2 column: 24 | **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 |
| post("HelloWorld!"); | No Error | No Error | **OK** |
| post(Name); | No Error | No Error | **OK** |
| post(1); | unexpected token "Numlit", expected one of "Stringlit", or "id", on line: 2 column: 6 | unexpected token "Numlit", expected one of "Stringlit", or "id", on line: 2 column: 6 | **OK** |
| post("One");  post("Two");  post("Three"); | No Error | No Error | **OK** |
| post("Hello") | unexpected token "}", expected ";", on line: 3 column: 1 | unexpected token "}", expected ";", on line: 3 column: 1 | **OK** |
| captured(#num); | 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** |
| PrimaryMission() {  captured(#num)  } deploy(); | unexpected token "}", expected ";", on line: 3 column: 1 | unexpected token "}", expected ";", on line: 3 column: 1 | **OK** |
| PrimaryMission() {  captured(1num);  } deploy(); | unexpected token "Numlit", expected "#", on line: 2 column: 10 | unexpected token "Numlit", expected "#", on line: 2 column: 10 | **OK** |
| captured(#num);  PrimaryMission() {  } deploy(); | 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** |
| PrimaryMission() {  } deploy();  captured(#num); | unexpected token "captured", expected <EOF>, on line: 3 column: 1 | unexpected token "captured", expected <EOF>, on line: 3 column: 1 | **OK** |
| PrimaryMission() {  captured(#num);  captured(#name);  captured(#addr);  } deploy(); | No Error | No Error | **OK** |

**Function Declaration**

Syntax:

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

|  |  |  |  |
| --- | --- | --- | --- |
| SAMPLE INPUT | SYSTEM Expected Output | SYSTEM Actual Output | Remarks |
| unit MissionOne(unit a, unit b, unit c) {  post("Mission1");  backup(a);  } | No Error | No Error | **OK** |
| unit MissionOne(unit a, unit b, unit c) {  post("Mission1");  backup(a);  }  unit Missiontwo(unit a) {  post("Mission2");  backup(a);  } | No Error | No Error | **OK** |
| miss MissionOne(unit a, unit b, unit c) {  post("Mission1");  backup(a);  }  unit Missiontwo(unit a) {  post("Mission2");  backup(a);  } | unexpected token "backup", expected "}", on line: 3 column: 1 | unexpected token "backup", expected "}", on line: 3 column: 1 | **OK** |
| miss MissionOne(unit a, unit b, unit c) {  post("MissionMiss");  } | No Error | No Error | **OK** |
| miss MissionOne(unit a, unit b, unit c) {  post("MissionMiss");  captured(#num);  } | No Error | No Error | **OK** |
| miss MissionOne(unit a, unit b, unit c) {  unit a=0;  joe a = 'd';  company="abortmission";  } | No Error | No Error | **OK** |
| miss MissionOne(unit a, unit ) {  unit a=0;  joe a = 'd';  company="abortmission";  } | 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** |
| miss MissionOne(unit a[] ) {  unit a=0;  joe a = 'd';  company="abortmission";  } | No Error | No Error | **OK** |
| miss MissionOne(unit a[] , unit c[] ) {  unit a=0;  joe a = 'd';  company="abortmission";  } | No Error | No Error | **OK** |
| unit b=0;  miss MissionOne(unit a[] , unit c[] ) {  unit a=0;  joe a = 'd';  company="abortmission";  } | No Error | No Error | **OK** |
| miss MissionOne(unit a[] , unit c[] ) {  unit a=0;  joe a = 'd';  company="abortmission";  }  unit b=0; | unexpected token "=", expected "(", on line: 6 column: 7 | unexpected token "=", expected "(", on line: 6 column: 7 | **OK** |

**Assignments**

Syntax:

<AccessAssignDtype> = id <N1> <operand>;

<AccessAssignDtype> = id = <operand>;

<AccessAssignmentsMath> = id <operEq> <MathOp;

<mntCond< = mnt id;

<scan> = captured (#id);

|  |  |  |  |
| --- | --- | --- | --- |
| SAMPLE INPUT | SYSTEM Expected Output | SYSTEM Actual Output | Remarks |
| PrimaryMission() {  num=num+num;  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  num=num+num\*num/num-num;  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  num=num^4;  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  num = 1;  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  i++;  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  ++i;  } deploy(); | 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** |
| PrimaryMission() {  +a;  } deploy(); | unexpected token "+", expected "}", on line: 2 column: 1 | unexpected token "+", expected "}", on line: 2 column: 1 | **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** |
| PrimaryMission() {  num=num=num=num=num=num;  } deploy(); | unexpected token ";", expected "}", on line: 2 column: 18 | unexpected token ";", expected "}", on line: 2 column: 18 | **OK** |

**Inquire statement**

Syntax:

<for> = inquire ( id = Numlit ; id <op1> Numlit ; <mntcond> ; ) { <statement> } <body>

|  |  |  |  |
| --- | --- | --- | --- |
| SAMPLE INPUT | SYSTEM Expected Output | SYSTEM Actual Output | Remarks |
| PrimaryMission() {  inquire(i=0;i<= 1;i++;) {  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inquire(i=m;i<= 1;i++; ) {  }  } deploy(); | 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** |
| PrimaryMission() {  inquire(i=0;i<=1; ++i; ) {  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inquire(i=0;i<=1; ++i; ) {  i++;  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inquire(i=0;i<=1; ++i; ) i++;  }  } deploy(); | unexpected token "id", expected "{", on line: 2 column: 36 | unexpected token "id", expected "{", on line: 2 column: 36 | **OK** |
| PrimaryMission() {  inquire(i=0;i<=1; ++i; ) {  num=num+1;  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inquire(i=0;i<=1; ++i; ) { {  num=num+1;  }  } deploy(); | unexpected token "{", expected "}", on line: 2 column: 37 | unexpected token "{", expected "}", on line: 2 column: 37 | **OK** |
| PrimaryMission() {  inquire(i=0;i<=1; ++m; ) {  num=num+1;  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inquire(i=0;i<=1; +i; ) {  }  } deploy(); | unexpected token "+", expected one of "++", "--", or "id", on line: 2 column: 30 | unexpected token "+", expected one of "++", "--", or "id", on line: 2 column: 30 | **OK** |
| PrimaryMission() {  inquire(i==0;i<=1; ++i; ) {  }  } deploy(); | unexpected token "=", expected one of "Numlit", or "Zero", on line: 2 column: 12 | unexpected token "=", expected one of "Numlit", or "Zero", on line: 2 column: 12 | **OK** |
| inquire(i=0;i<=1; ++i; ) {  }  PrimaryMission() {  } deploy(); | unexpected token "inquire", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1 | unexpected token "inquire", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1 | **OK** |
| PrimaryMission() {  } deploy();  inquire(i=0;i<=1; ++i; ) {  } | unexpected token "inquire", expected <EOF>, on line: 3 column: 1 | unexpected token "inquire", expected <EOF>, on line: 3 column: 1 | **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 |
| PrimaryMission() {  inorder(i<=0) {  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder(i<=0 || i>0) {  }  } deploy(); | unexpected token "||", expected ")", on line: 2 column: 19 | unexpected token "||", expected ")", on line: 2 column: 19 | **OK** |
| PrimaryMission() {  inorder((i<=0) || (i>0)) {  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder((i<=0) || (i>0) {  }  } deploy(); | unexpected token "{", expected ")", on line: 2 column: 34 | unexpected token "{", expected ")", on line: 2 column: 34 | **OK** |
| PrimaryMission() {  inorder((i<=0) & (i>0)) {  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder((i<=0) || (i==0) & (b>5) & ) {  }  } deploy(); | unexpected token ")", expected "(", on line: 2 column: 48 | unexpected token ")", expected "(", on line: 2 column: 48 | **OK** |
| PrimaryMission() {  inorder((i<=0) || (i==0) & (b>5) & (m!0)) {  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder(((((((i<=0) || (i==0) & (b>5) & (m!0)) {  }  } deploy(); | unexpected token "(", expected one of "Numlit", "Declit", "id", "Stringlit", or "joe", on line: 2 column: 10 | unexpected token "(", expected one of "Numlit", "Declit", "id", "Stringlit", or "joe", on line: 2 column: 10 | **OK** |
| PrimaryMission() {  inorder((i<=0) || (i==0) & (b>5) & (m!0)) {  post("ifelse");  }  } deploy(); | No Error | No Error | **OK** |
| inorder((i<=0) || (i==0) & (b>5) & (m!0)) {  post("ifelse");  }  PrimaryMission() {  } deploy(); | unexpected token "inorder", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1 | unexpected token "inorder", expected one of "hold", "unit", "digit", "company", "joe", "response", "miss", "struct", or "PrimaryMission", on line: 1 column: 1 | **OK** |
| PrimaryMission() {  } deploy();  inorder((i<=0) || (i==0) & (b>5) & (m!0)) {  post("ifelse");  } | unexpected token "inorder", expected <EOF>, on line: 3 column: 1 | unexpected token "inorder", expected <EOF>, on line: 3 column: 1 | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder(j!=10) {  post("elseif2");  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  otherorder(j!=10) {  post("elseif2");  }  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  } deploy(); | unexpected token "otherorder", expected "}", on line: 2 column: 1 | unexpected token "otherorder", expected "}", on line: 2 column: 1 | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder(j!=m || ) {  post("elseif2");  }  } deploy(); | unexpected token "||", expected ")", on line: 8 column: 18 | unexpected token "||", expected ")", on line: 8 column: 18 | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder((j!=m) || (l<0)) {  post("elseif2");  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder((j!=m) || (l<0)) {  post("elseif2");  }  order {  post("else");  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder((j!=m) || (l<0)) {  post("elseif2");  }  order (i=3) {  post("else");  }  } deploy(); | unexpected token "(", expected "{", on line: 11 column: 6 | unexpected token "(", expected "{", on line: 11 column: 6 | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder((j!=m) || (l<0)) {  post("elseif2");  }  } deploy();  order {  post("else");  } | unexpected token "order", expected <EOF>, on line: 12 column: 1 | unexpected token "order", expected <EOF>, on line: 12 column: 1 | **OK** |
| PrimaryMission() {  order {  post("else");  }  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder((j!=m) || (l<0)) {  post("elseif2");  }  } deploy(); | unexpected token "order", expected "}", on line: 2 column: 1 | unexpected token "order", expected "}", on line: 2 column: 1 | **OK** |
| PrimaryMission() {  inorder(i<=0) {  post("ifelse");  }  otherorder(i>=5) {  post("elseif");  }  otherorder((j!=m) || (l<0)) {  post("elseif2");  }  order  post("else");  }  } deploy(); | unexpected token "post", expected "{", on line: 12 column: 1 | unexpected token "post", expected "{", on line: 12 column: 1 | **OK** |

**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 |
| PrimaryMission() {  go {  } phase(a!=0);  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  go {  unit a=0;  response isGo=AFFIRMATIVE;  post("go");  } phase(a!=0);  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  go {  unit a=0;  response isGo=AFFIRMATIVE;  } phase(a!=0);  } deploy(); | 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** |
| PrimaryMission() {  go {  post("GoFirst");  go {  post("GoSecond");  } phase(b>=0);  } phase(a!=0);  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  go {  post("GoFirst");  go {  post("GoSecond");  } phase(a!=0);  } deploy();  } phase(b>=0); | unexpected token "deploy", expected "phase", on line: 7 column: 2 | unexpected token "deploy", expected "phase", on line: 7 column: 2 | **OK** |
| PrimaryMission() {  go {  post("GoFirst");  } phase(b>=0);  go {  post("GoSecond");  } phase(a!=0);  } deploy(); | No Error | No Error | **OK** |
| } phase(b>=0);  PrimaryMission() {  go {  post("GoFirst");  go {  post("GoSecond");  } phase(a!=0);  } deploy(); | 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** |
| PrimaryMission() {  phase(i<90) {  post("Average");  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  phase(i<90) {  post("Average");  phase(i>90) {  post("Low");  }  }  } deploy(); | 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 |
| PrimaryMission() {  campaign(num) {  operation a:  post("operation1");  abort();  operation b:  post("operation2");  abort();  }  } deploy(); | No Error | No Error | **OK** |
| PrimaryMission() {  campaign(num) {  operation a:  post("operation1");  abort();  operation b:  post("operation2");  }  } deploy(); | unexpected token "}", expected "abort", on line: 8 column: 1 | unexpected token "}", expected "abort", on line: 8 column: 1 | **OK** |
| PrimaryMission() {  operation a:  post("operation1");  abort();  campaign(num) {  operation b:  post("operation2");  abort();  }  } deploy(); | unexpected token "operation", expected "}", on line: 2 column: 1 | unexpected token "operation", expected "}", on line: 2 column: 1 | **OK** |
| PrimaryMission() {  } deploy();  campaign(num) {  operation a:  post("operation1");  abort();  operation b:  post("operation2");  abort();  } | unexpected token "campaign", expected <EOF>, on line: 3 column: 1 | unexpected token "campaign", expected <EOF>, on line: 3 column: 1 | **OK** |
| PrimaryMission() {  campaign(num) {  operation a:  post("operation1");  campaign(num2) {  operation ab:  post("operation1.5");  abort();  }  abort();  operation b:  post("operation2");  abort();  }  } deploy(); | No Error | No Error | **OK** |