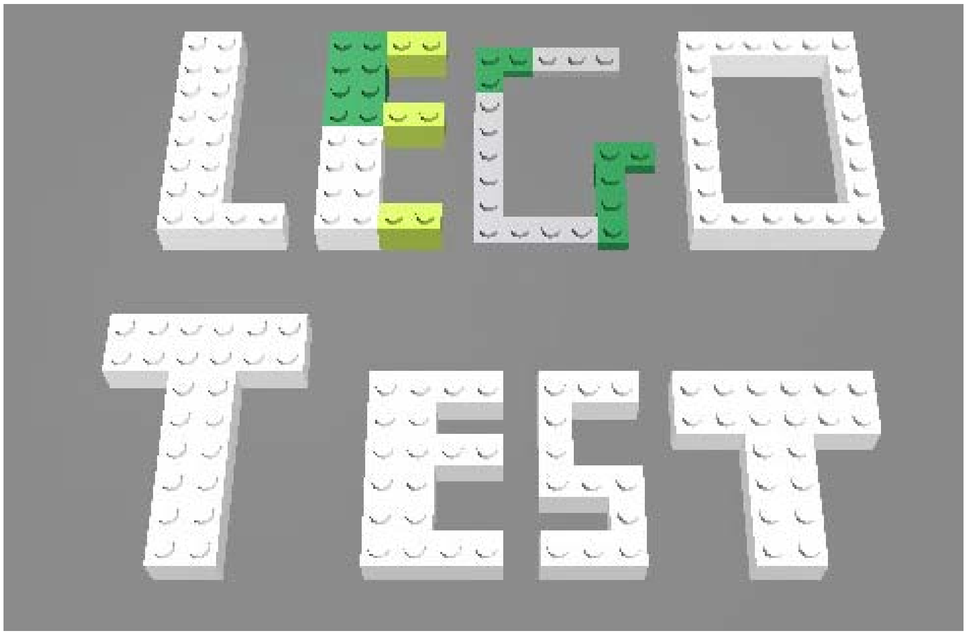


Robotics Unit Exam

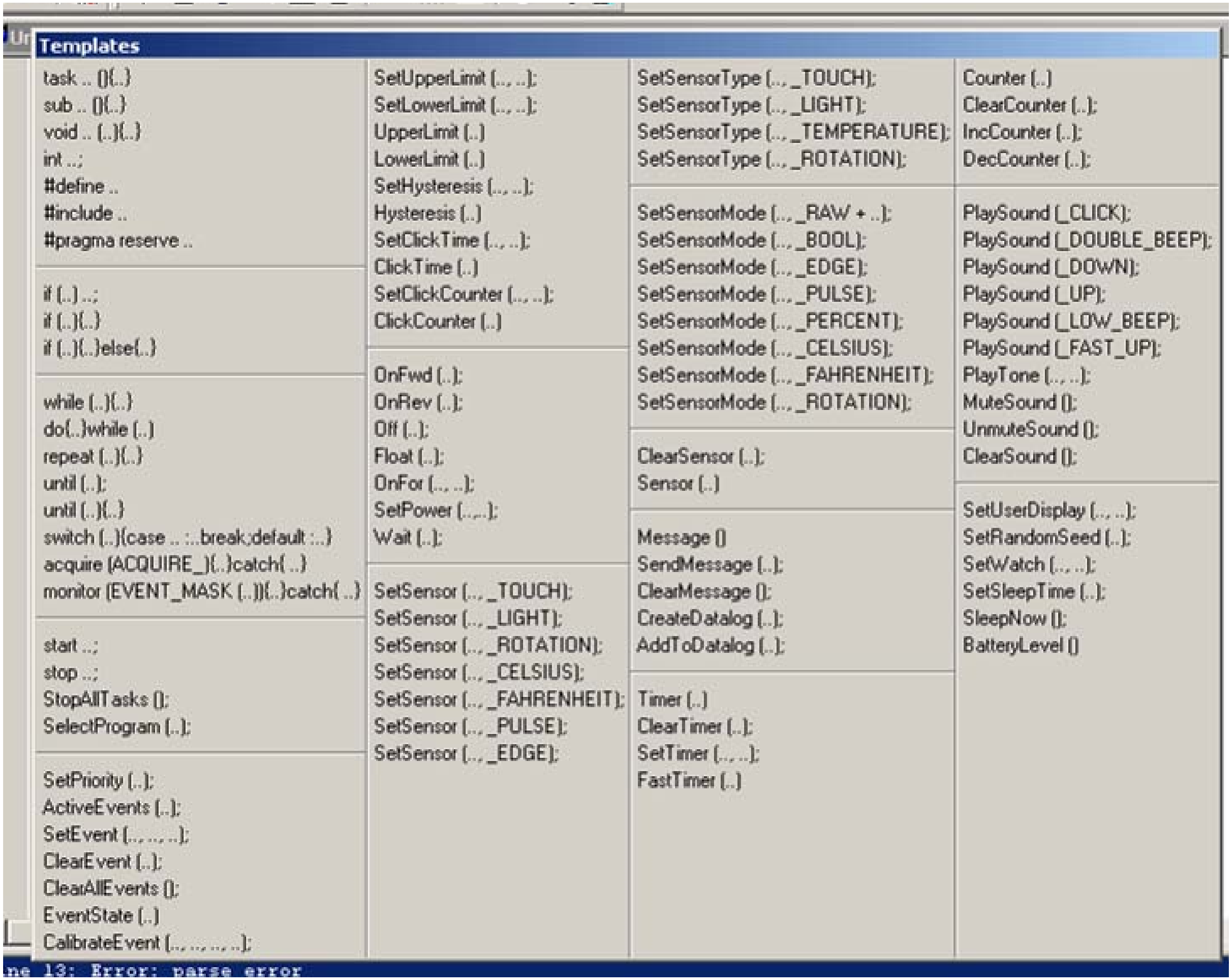
Excel High School
Pre-Engineering

Name _solution___________

Date ______________

Teacher ___________

The command list for your reference:



```
task .. (){..}
sub .. (){..}
void .. (..){..}
int ..;
#define ..
#include ..
#pragma reserve ..

if (..) ..;
if (..){..}
if (..){..}else{..}

while (..){..}
do{..}while (..)
repeat (..){..}
until (..);
until (..){..}
switch (..){case .. :..break;default :..}
acquire (ACQUIRE_){..}catch{..}
monitor (EVENT_MASK (..)){..}catch{..}

start ..;
stop ..;
StopAllTasks ();
SelectProgram (..);

SetPriority (..);
ActiveEvents (..);
SetEvent (.., .., ..);
ClearEvent (..);
ClearAllEvents ();
EventState (..)
CalibrateEvent (.., .., .., ..);
```

```
SetUpperLimit (.., ..);
SetLowerLimit (.., ..);
UpperLimit (..)
LowerLimit (..)
SetHysteresis (.., ..);
Hysteresis (..)
SetClickTime (.., ..);
ClickTime (..)
SetClickCounter (.., ..);
ClickCounter (..)

OnFwd (..);
OnRev (..);
Off (..);
Float (..);
OnFor (.., ..);
SetPower (..,..);
Wait (..);

SetSensor (.., _TOUCH);
SetSensor (.., _LIGHT);
SetSensor (.., _ROTATION);
SetSensor (.., _CELSIUS);
SetSensor (.., _FAHRENHEIT);
SetSensor (.., _PULSE);
SetSensor (.., _EDGE);
```

```
SetSensorType (.., _TOUCH);
SetSensorType (.., _LIGHT);
SetSensorType (.., _TEMPERATURE);
SetSensorType (.., _ROTATION);

SetSensorMode (.., _RAW + ..);
SetSensorMode (.., _BOOL);
SetSensorMode (.., _EDGE);
SetSensorMode (.., _PULSE);
SetSensorMode (.., _PERCENT);
SetSensorMode (.., _CELSIUS);
SetSensorMode (.., _FAHRENHEIT);
SetSensorMode (.., _ROTATION);

ClearSensor (..);
Sensor (..)

Message ()
SendMessage (..);
ClearMessage ();
CreateDatalog (..);
AddToDatalog (..);

Timer (..)
ClearTimer (..);
SetTimer (.., ..);
FastTimer (..)
```

```
Counter (..)
ClearCounter (..);
IncCounter (..);
DecCounter (..);

PlaySound (_CLICK);
PlaySound (_DOUBLE_BEEP);
PlaySound (_DOWN);
PlaySound (_UP);
PlaySound (_LOW_BEEP);
PlaySound (_FAST_UP);
PlayTone (.., ..);
MuteSound ();
UnmuteSound ();
ClearSound ();

SetUserDisplay (.., ..);
SetRandomSeed (..);
SetWatch (.., ..);
SetSleepTime (..);
SleepNow ();
BatteryLevel ()
```

```
ne 13: Error: parse error
```

1) Answer the following questions:

a) What is a subroutine and how is it used?

A smaller part (sub) of a program. Something that may be done repeatedly (routine) and can be simplified by defining what it is that is done once, and reusing this definition.

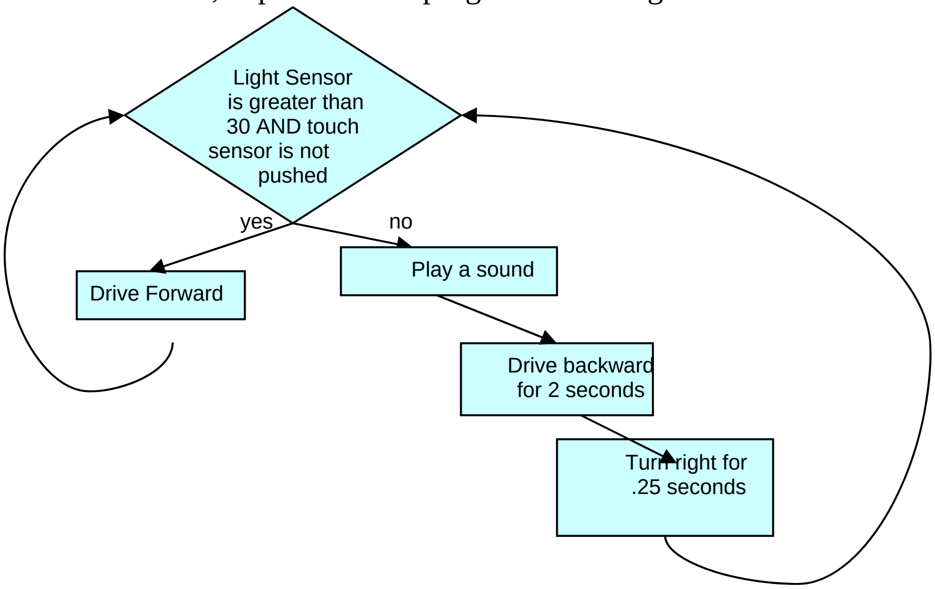
b) What is the benefit of using a subroutine?

Simplifies a program where a particular set of commands may be used more than one times under varying circumstances

c) Give an example of everyday life where a subroutine could be useful if you could "program" your life.

Opening a door: Opening a door would consist of something like " lift right arm until touches handle, open fingers, move hand forward, close fingers, twist hand…". Each time I want to open a door, I just have to tell myself "open door" instead of thinking of each step

2) In 3-4 sentences, explain what a program following the below flowchart will do:


Light Sensor is greater than 30 AND touch sensor is not pushed
yes
no
Drive Forward
Play a sound
Drive backward for 2 seconds
Turn right for .25 seconds

Write your explanation below:

If the light sensor reads a value greater (but not equal) to 30, AND the touch sensor has not bumped into anything, the bot will keep driving forward. As soon as EITHER the light value drops below 30 OR the touch sensor is bumped, the bot will play a sound, then backup for 2 seconds, turn right for ¼ seconds (I assume this equates to 90 degrees). At this point in time, the process will repeat.

3) **Convert the flowchart of question #2 into a program using the NQC programming language (what we used in Bricx Command Center). There are many correct answers to this problem! Don’t worry about programming “grammar” errors such as incorrect capitalization. Remember that you may use the command reference sheet.**

a) Draw a diagram of your control brick. Clearly label:

- Inputs/sensors light sensor on sensor 1 port, touch sensor on port 2
- Outputs/motors left motor on A, right motor on C
- Direction of travel forward implies that motors A and C are rotating in the “forward” direction

b) write your program in the space below:

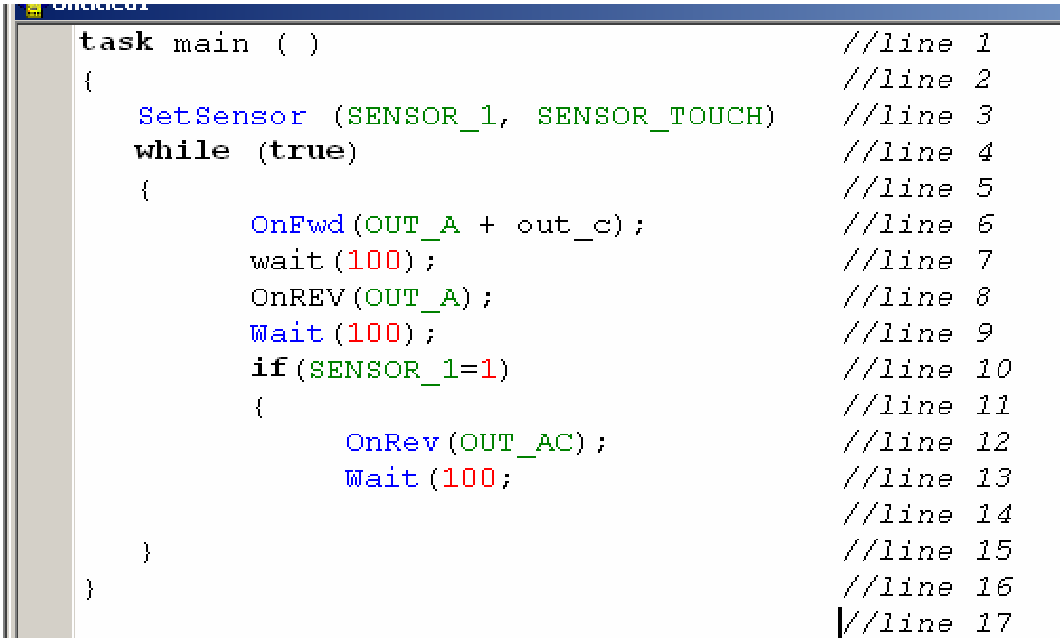
```
task main( )
{
        SetSensor(SENSOR_1, TYPE_LIGHT);
        SetSensor(SENSOR_2, TYPE_TOUCH);
        while(true)
        {
        `       if (SENSOR_1>30 && SENSOR_2==0)
                {
                        OnFwd(OUT_AC);
                }
                else
                {
                        PlaySound(0);
                        OnRev(OUT_AC);
                        OnFwd(OUT_A);
                        Wait(25);
                }
        }




}
```

4) Considering the task from questions 2 and 3, explain how you would construct your robot. Be specific about size, wheels, and attachments you would use. Be sure to include a labeled diagram.

5) Take a look at the following program and identify a solution to each of the errors indicated by the compiler. Remember that sometimes the compiler doesn't find an error until it is on a line after the error.



```
task main ( )                                       //line 1
{                                                   //line 2
    SetSensor (SENSOR_1, SENSOR_TOUCH)              //line 3
    while (true)                                    //line 4
    {                                               //line 5
            OnFwd(OUT_A + out_c);                   //line 6
            wait(100);                              //line 7
            OnREV(OUT_A);                           //line 8
            Wait(100);                              //line 9
            if(SENSOR_1=1)                          //line 10
            {                                       //line 11
                  OnRev(OUT_AC);                    //line 12
                  Wait(100;                         //line 13
                                                    //line 14
    }                                               //line 15
}                                                   //line 16
                                                    //line 17
```

a)

Line 3: Error: parse error

Solution? Add “;”

Line 6: Error: undefined variable ‘out_c’

Solution? Should be “OUT_C”

Line 7: Error: undefined function ‘wait’

Solution? Should be “Wait”

Line 8: Error: undefined function ‘OnREV’

Solution? Should be “OnRev”

Line 10: Error: parse error

Solution? ‘= =”

Line 13: Error: parse error

Solution? 100)

Line 17: Error: parse error

Solution? Missing “}”

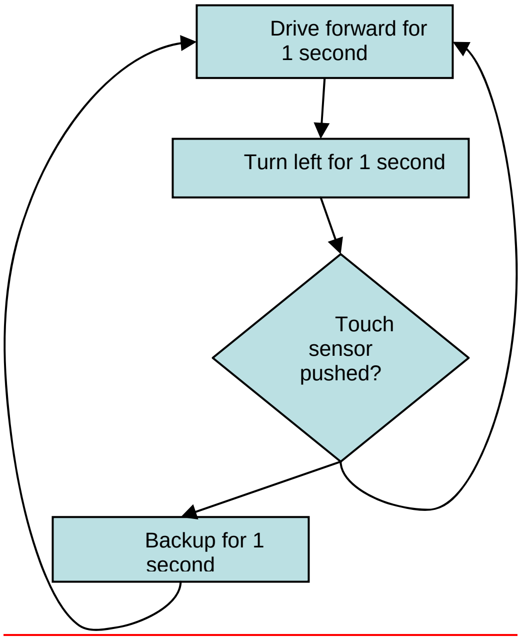
b) All of these errors are examples of **`syntax`** or **`logical`** errors? (circle one)

^^^^^^

c) Assuming that all the identified errors are corrected, the touch sensor is on the front of the robot, Motor A is on the left and Motor C is on the right, explain in 2-3 sentences what this program will do.

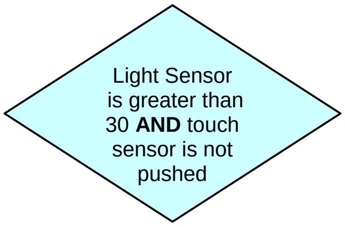
The bot will drive forward for 1 second and turn left for 1 second. After this, if the touch sensor is pushed, then the bot will backup for 1 second and then repeat the sequence. However, if the touch sensor is **not** pushed, then the bot will then simply repeat the sequence with out the 1 second backup.

d) With the same assumptions from c), make a flowchart to illustrate what the program will make the robot do.



6) Based on some possible combinations of values for a light and touch sensor in the chart below, complete the table for the decision making of the flow chart element to the right.

Light sensor reading	Touch sensor reading	YES or NO?
25 is not >30, 0	0, yes, it is not pushed, 1	0*1=0/no/false
30 is not >30, 0	1, no, it is pushed, 0	0*0=0/no/false
45 is >30, 1	0, yes, it is not pushed, 1	1*1=1/yes,/true
40 is >30, 1	1, no, it is pushed, 0	1*0=0/no/false
30 is not >30, 0	0, yes, it is not pushed, 1	0*1=0/no/false
15 is not >30, 0	1, no, it is pushed, 0	0*0=0/no/false


Light Sensor is greater than 30 AND touch sensor is not pushed

This is an example of AND logic

light	**Touch**	**Result = light*touch**
0	**0**	**0*0=0**
0	**1**	**0*1=0**
1	**0**	**1*0=0**
1	**1**	**1*1=1**

7) In 2-3 sentences, explain the difference between a **syntax error** and a **logic error.**

Syntax error: a language or "grammatical" error where the program doesn't understand what you are telling it to do. For instance if my native language is not English and I am telling you to do something in English by writing "stEPS dhye to ad agwe" you wonldn't have any idea what I am saying.

Logic error: the compiler understand the way you tell it to do something, but the end result is that program does not do what you want it to do. This means that you programmed your robot wrong. With the same example, as before, this is as if I want to tell you to walk forward 2 steps. But because my English is not very good, I say "step backwards 1 step." You will understand what I say, but not do what I mean for you to do.

a) Which of these two error types can the compiler **only** detect and why?

Syntax because the compiler only knows proper "programming grammar" but it doesn't know how to use it. In other words, the compiler can only check if it can understand you, but it has no way of testing if you are telling it to the right thing.

8) In 2-3 sentences, explain how the light sensor works.

When the light detector sees light, it assigns a value between 0 and 100 to the light where 0 is darkest and 100 is brightest. Light can be reflected into the sensor or just be stray light.

a) Why will the light sensor still work even if the room is completely dark?

Because it has its own light source which is the red glowing light. The red light reflects off surfaces indicating the type of surface &/or how far away the surface is.

b) From classroom discussions, list 4 variables that affect how well the light sensor operates. (there are more than 4)

i) Distance from sensor to surface
ii) color of surface
iii) surface finish (rough, glossy, etc)
iv) stray background light

9) You want a robot to drive forward for 10 seconds, turn off and play a sound. Write the code for this task: Same assumptions as in #5c

```
task main( )
{
        OnFwd(OUT_AC)
        Wait(1000);
        Off(OUT_AC);
        PlaySound(0);
}
```