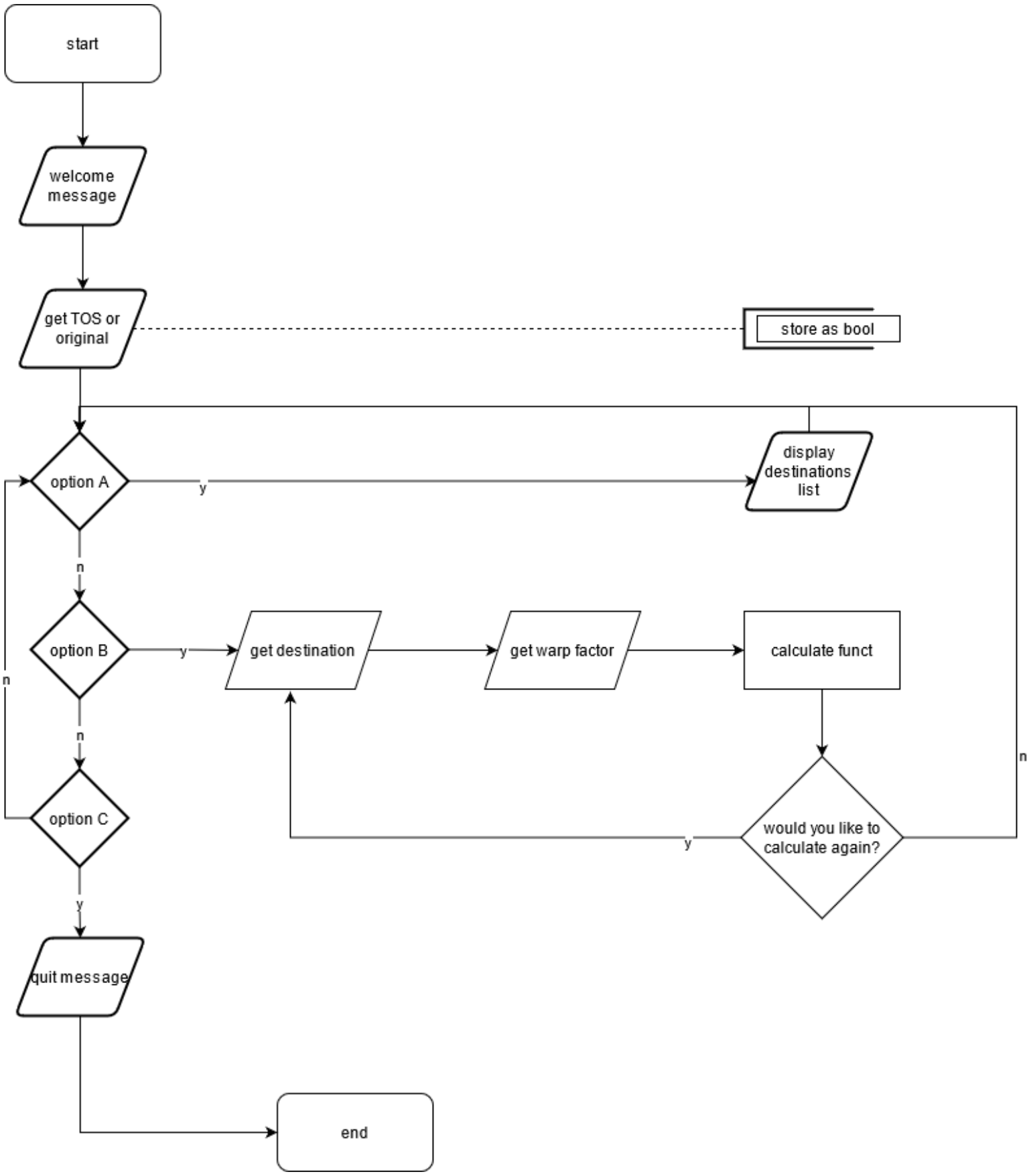
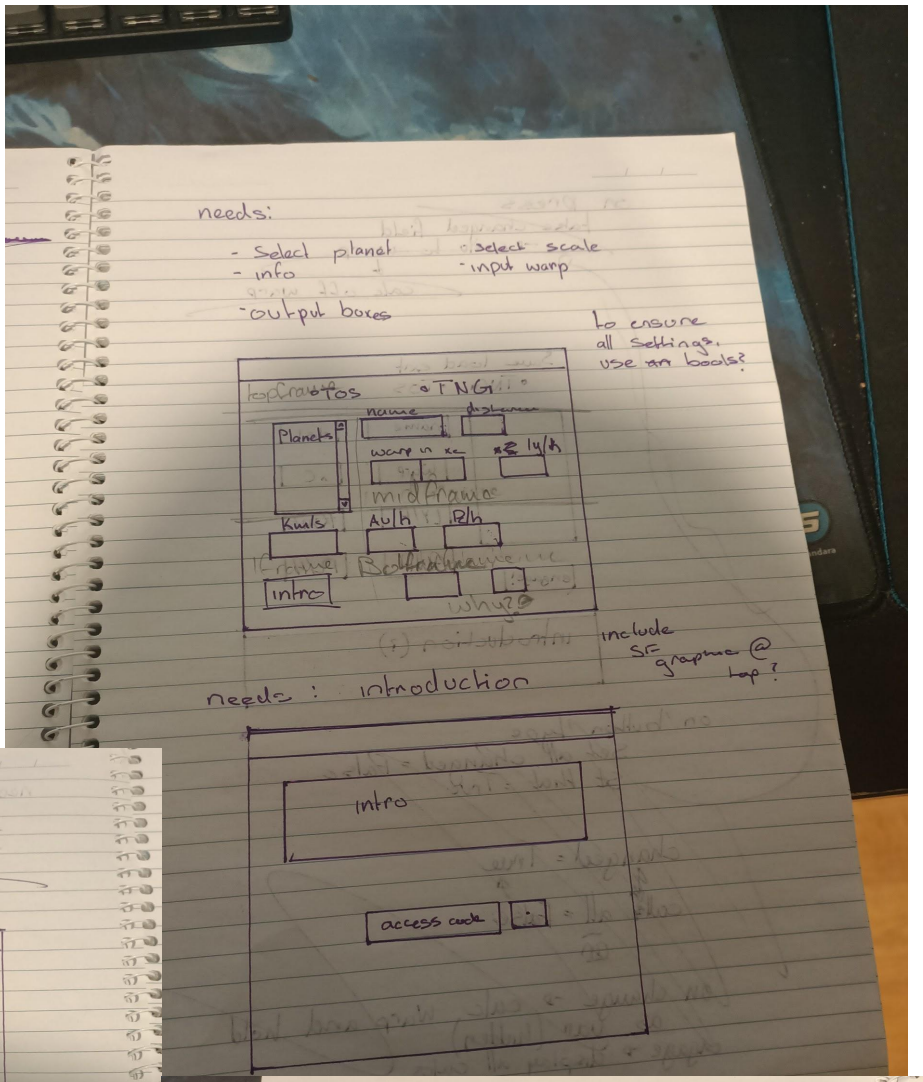
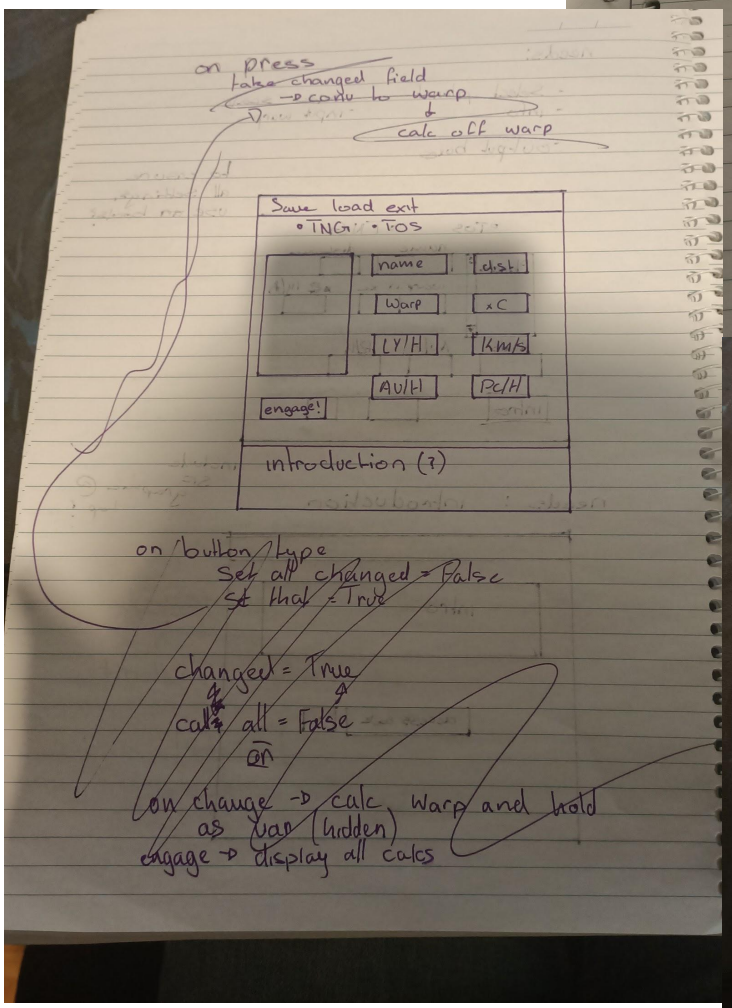
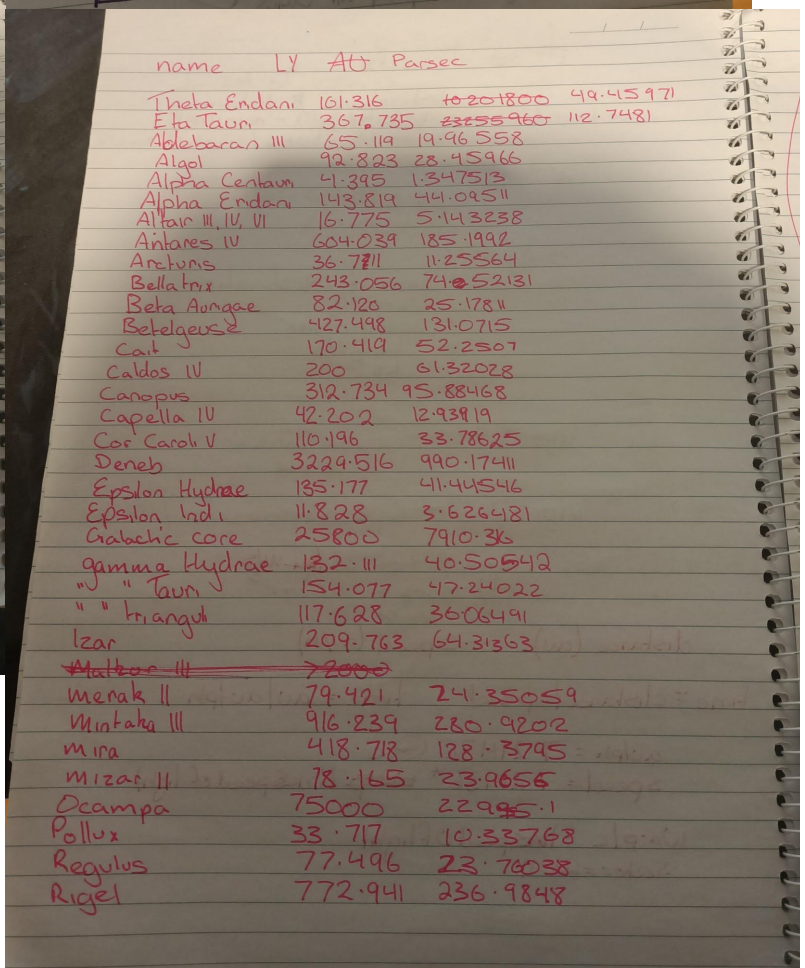
Flowchart


start
welcome message
get TOS or original
store as bool
option A
y
display destinations list
n
option B
y
get destination
get warp factor
calculate funct
n
n
option C
would you like to calculate again?
y
n
y
quit message
end

Hand Calculations







name	LY	AU	Parsec
Theta Endani	161.316	~~10201800~~	49.45971
Eta Tauri	367.735	~~23255960~~	112.7481
Aldebaran III	65.119	19.96558	
Algol	92.823	28.45966	
Alpha Centauri	4.395	1.347513	
Alpha Endani	143.819	44.09511	
Altair III, IV, VI	16.775	5.143238	
Antares IV	604.039	185.1992	
Arcturis	36.711	11.25564	
Bellatrix	243.056	74.52131	
Beta Aurigae	82.120	25.1781	
Betelgeuse	427.498	131.0715	
Cait	170.419	52.2507	
Caldos IV	200	61.32028	
Canopus	312.734	95.88468	
Capella IV	42.202	12.93919	
Cor Caroli V	110.196	33.78625	
Deneb	3229.516	990.1741	
Epsilon Hydrae	135.177	41.44546	
Epsilon Indi	11.828	3.626481	
Galactic core	25800	7910.36	
gamma Hydrae	132.111	40.50542	
" " Tauri	154.077	47.24022	
" " triangul	117.628	36.06491	
Izar	209.763	64.31363	
~~Markab III~~	~~>2000~~		
Menak II	79.421	24.35059	
Mintaka III	916.239	280.9202	
mira	418.718	128.3795	
Mizar II	78.165	23.9656	
Ocampa	75000	22995.1	
Pollux	33.717	10.33768	
Regulus	77.496	23.76038	
Rigel	772.941	236.9848	

Initial Pseudo-Code

```
Program: Main
Start
Print (welcome message)
If [scaleSet = false]
        print(Would you like to use the original series (1) or the next generation (2) warp
        scale
        get(answer)
                if[answer=1]
                        Set scaleSet = true
                        Set scaleType = 0
                        Print (thankyou! You are now using the Original Series warp scale)
                if[answer=2]
                        Set scaleSet = true
                        Set scaleType = 1
                        Print (thankyou! You are now using the Next Generation warp scale)
                Else
                        print(please select 1 or 2)
Print (menu)
If [Option A]
        Print(possible destinations)
        Print %Destinations%
ElseIf [Option B]
        Print(Please input a destination)
        Print %destinations%
        get(destination)
        Check(destination)

        print(destination accepted… please input warp factor)
        get(warp factor)
        calculate(destination, warpfactor, scaleType)

        print(Would you like to calculate again?)
        Get(response)
        If [y]
                Goto menu
        If [n]
                quit()
ElseIf [Option C]
        Print (would you like to quit?)
        If [n]
                Goto Menu
        If [y]
                quit()
Else loop

quit()
```

```
Print(live long and prosper)
End

Function: calculate(destination, warpfactor, scaleType):
        If [scaletype = 0]
                warpC = Warpfactor^3  #warp speed, factor of speed of light
        If [scaletype = 1]
                warpC = warpfactor^10/3          #warp speed, factor of speed of light
destinationName = [destination]
destinationDistance = [destination]+1
        #list will be [destination, distance, destination, distance] etc….
        #these should find and match the given destination in the list
warpTimesC = warpC/299792.458 #warp speed in meters per second
                warpAUH = 7.21436*warpC
                warpPCH = warpAUH*0.000004848137
                Kms = warpC*299792.458
                timeToDestination = destinationDistance/warpAUH

print(warp factor:)
print(%Warpfactor%)
print(speed relative to C)
print(%warpC% times the speed of light)
print(AU per hour)
print(%warpAUH%)
print(KM/s)
print(%Kms%)
print(distance)
print(%destinationDistance%)
print(Travel time to %destinationName%)
print(%timeToDestination%)

Function: check(destination)
Checks if destination is in the list

https://www.stdimension.org/int/Cartography/DistanceList.htm
```

I - Background

You are the helmsman aboard the *USS Animus* (NCC-74293), a United Federation of Planets *Valiant* class destroyer. After shore leave on Earth, the *Animus* is ready to warp to a destination to fulfil whatever mission she's been assigned. With this tool, you will be able to calculate time to any one of several destinations in the navigation computer, along with relative speeds and distance.

Distances are in Parsecs, and speeds are displayed in Parsecs/Hour, Astronomical Units/Hour and Kilometers/Second.

II - Business rules

a) The problem at hand
   We need to calculate the possible speeds at which the *Valiant* will be able to reach any given planet from her starting point in sector 001 (the Sol system). This includes the normal cruising speed, maximum cruising speed and maximum emergency speed - warps 8.5, 9.8 and 9.96 **(forty hours maximum)** respectively. We need to be able to calculate the speed in different units, and need to be able to calculate warp speeds of different scales, being the Original scale and Re-vamped scale.
b) Inputs will be:
   i) Destination
   ii) Warp Scale type (original, revamped)

Bibliography

LCARSCom.Net | The LCARS Computer Network | A Star Trek Fan Site. 2021. *sounds*. [online] Available at: <https://www.lcarscom.net/sounds/> [Accessed 2 May 2021].

Mediacollege.com. 2021. *Star Trek: The Original Series Sound Effects*. [online] Available at: <https://www.mediacollege.com/downloads/sound-effects/star-trek/tos/> [Accessed 2 May 2021].