**Inputs**

* 3 integer values from keyboard

**Outputs**

* Message asking for user to input 3 integer values
* Message displaying final CPU answer
* Message displaying final FPU answer
* Message asking if user wants to continue program or exit

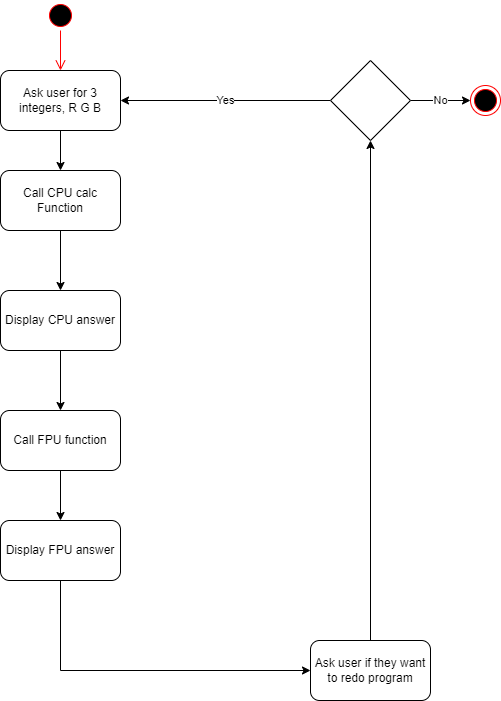
**Variables**

* Output – BYTE for saying display function final CPU answer: msgCPU
* Output – BYTE for saying display function final FPU answer: msgFPU
* Output – BYTE message for input integer of B: msgB
* Output – BYTE message for input integer of G: msgG
* Output – BYTE message for input integer of R: msgR
* Output – BYTE for asking user if they want to exit program: msgExit

**Algorithm**

1. Ask user to input 3 strings (R,G,B)
2. Call a function to get the CPU calculation [R \* 3 / 10 + G \* 6 / 10 + B \* 1 / 10]
3. Display CPU answer
4. Call a function to get the FPU calculation [R \* 0.299 + G \* 0.587 + B \* 0.114]
5. Display FPU answer
6. Ask user if they want to exit or repeat program

**Flow Diagram**



**Stack Diagrams**

**GrayScaleCPU Function**

|  |
| --- |
|  |
| B |
| G |
| R |
| return address |
| Old EBP |
| 3/10 |
| 6/10 |
| Final CPU answer |
| EBX |
| ECX |
| EDX |
| FLAGS <-ESP |
|  |

**greyScaleFPU function**

|  |
| --- |
|  |
| B |
| G |
| R |
| return address |
| Old EBP |
| 114 |
| 587 |
| 229 |
| B\*114 |
| G\*587 |
| R\*229 |
| Final answer |
| EAX |
| EBX |
| ECX |
| EDX |
| FLAGS <-ESP |
|  |

**Main function**

|  |
| --- |
| return address |
| Old EBP |
| R |
| G |
| B |
|  |