the program is in [X] user [ ] kernel mode. (**Q10**, please answer the state of the process, and choose the right mode from the above options)

```
      total1 = (t1+t2+t3)*0.1+Project_total*0.50 +
Homework_total*0.1 + Exam_total*0.3);
```

-- **Event:** interrupted due to time quantum expiration →**Q11**, the state of the process will be changed from ___**running**___state to ___**ready**___state, before the transition, the CPU will handle the timeout and switch to execute another process (context switch)

-- **Event:** scheduled to run →**Q12**the state of process will be changed from ___**ready**___state to __**running**__state(context switch)

```
      total2 = scale_the_grade(Project_total, Homework_total, m1, m2,
m3);
      if (total1 > total2) total = total1;
      else total = total2;
```

-- **Event:** I/O device interrupts to signal the completion of an I/O of a different process

→ Switch to [ ] user [X] kernel mode to run interrupt handler (context switch) (**Q13**, please check the correct one from the above options). **Q14**, the current process is in __**ready**__ state.

→ Switch back to run this process in [X] user [ ] kernel mode(context switch)**Q15**

```
      if (extra = 0) final = total;
      else final = (float) Bonus(total, extra);
        printf (" %.1f ", final);
        scanf ("%c", &ch); printf ("\n");
    }
    fork ();
```

-- **Event:** system call

→ Kernel trap, switch to kernel mode to handle fork (context switch)

Q16, OS creates the child process, put it in __**ready**__ queue

}