Assesment-3(Wednesday)

1.
```
#include <stdio.h>
#include <stdlib.h>

#define MAX_USERS 500000

void amazon_platform() {
int users;
if (users > MAX_USERS) {
   printf("System Overload! The platform has crashed due to excessive load.\n");

} else {
   printf("Platform is running smoothly with %d users.\n", users);
}
}

int main() {
   int users;

   printf("Enter the number of concurrent users: ");
   scanf("%d", &users);

   amazon_platform(users);

   return 0;
}
```

Output:
Enter the number of concurrent users: 450000
Platform is running smoothly with 30998 users.

2.
```
#include <stdio.h>

#define Total_recommendations 100
#define Failure_percent 2

int main() {
   float failed_recommendations = 0;

   for (int i = 1; i <= Total_recommendations; i++) {
      if (i % (100 / Failure_percent) == 0) {
         failed_recommendations++;
```

```
        }
    }

    printf("Total recommendations: %d\n", Total_recommendations);

    printf("Failed recommendations: %f\n", failed_recommendations);


    return 0;
}
```

Output:
Total recommendations: 100
Failed recommendations: 2.000000

5.
```
#include <stdio.h>

int main() {
    double debt = 1000000;
    double reduction = 0.05;
    while (debt > 1000) {
        printf("Debt: %.2f\n", debt);
        debt -= debt * reduction;
    }

    printf("Final Debt: %.2f\n", debt);
    return 0;
}
```

6.
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define STAGES 5

int main() {
    char *stages[STAGES] = {"Order Receipt", "Inventory Allocation", "Packaging", "Shipping",
"Delivery" };
     int time[STAGES];
     int optimizedTime[STAGES];
     int total = 0, optimizedTotal = 0;
```

Assesment-3(Wednesday)

```
    printf("Stage\t\tTime (s)\tOptimized (s)\n");

    for (int i = 0; i < STAGES; i++) {
        time[i] = rand() % 11 + 5;
        optimizedTime[i] = time[i] * 0.8;
        total += time[i];
        optimizedTotal += optimizedTime[i];

        printf("%-18s %5d %15d\n", stages[i], time[i], optimizedTime[i]);
    }

    printf("\nTotal Time: %d sec\n", total);
    printf("Optimized Time: %d sec\n", optimizedTotal);

    return 0;
}
```

Output:

```
Stage              Time (s)Optimized (s)
Order Receipt       11             8
Inventory Allocation   15          12
Packaging           11             8
Shipping            7              5
Delivery            6              4

Total Time: 50 sec
Optimized Time: 37 sec
```