MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG301 Algorithms and Data Structures - Worksheet 8

Structures

1. Write a program that includes a structure called Date as follows:

   ```
    struct Date{
   int month;
    int day;
    int year;
   }
   ```

   Define this structure global such that in can be used throughout the code. Write a program that takes the birthday of a user and stores in a variable which is struct Date type. Then process this birthday, and display the month in text.
   A sample run would be as follows:

   ```
   Enter your birthday: 1/1/2019
   You are born in January!
   ```

2. Update the program in part (1) such that it includes a function named days() that determines the number of days from the date 1/1/1900 to any birthday given to this function.
   Use the following date structure:

   ```
    struct Date{
   int month;
   int day;
   int year;
   }
   ```

   In writing the days function, use the convention that all years have 365 days and each month consist of 30 days. The function should return the number of days for any date passed to it.
   A sample output would be as follows:

   ```
   Enter your birthday: 1/1/2019
   You were born 43435 days later than 1/1/1900!
   ```

3. Update your program in part (2) such that it also has a new function named larger() that takes two dates in the structure given in part (a) and returns the larger date.
   A sample output would be as follows:

   ```
   Enter date 1: 10/9/2019
   Enter date 2: 12/4/2019
   12/9/2019
   ```

4. Explain what does the following program do:

   ```
    #include<stdio.h>
    #include<string.h>
    struct Student
    { int id;
    char name[32];
    };
    int main()
   { struct Student s;
    s.id = 101;
   strcpy(s.name, "John");
   printf("s.id = %d\n", s.id);
    printf("s.name = %s\n", s.name);
   printf("Please input id and name\n");
   ```

```
 scanf("%d", &s.id); scanf("%s", s.name);
printf("s.id = %d\n", s.id);
printf("s.name = %s\n", s.name);
return 0; }
```

5. Show the output of the following program.

```
#include<stdio.h>
#define EOD -1
 struct Student
 { int id;
 char name[32];
 double gpa; };
 void func1(int n, struct Student s[])
{ /* argument as an array */
 int i;
for(i=0; iid != EOD)
{ /* treat s as a pointer */
 printf("id = %d ", s->id);
 printf("name = %5s ", s->name);
printf("GPA = %.2f\n", s->gpa);
 s++; }
}
void func2(int n, struct Student *sp) /* argument as a pointer */
{  int i;
 for(i=0; iid != EOD)
{ /* treat sp as a pointer */
printf("id = %d ", sp->id);
 printf("name = %5s ", sp->name);
printf("GPA = %.2f\n", sp->gpa); sp++; }
}
 double averageGPA(struct Student *sp)
 { double gpa, totalval; int i; totalval=0.0; /* initialize the totalval */
for(i=0; sp[i].id != EOD; i++)
{ totalval += sp[i].gpa; /* add a GPA to totalval */
}
gpa = totalval/(double)i; /* calculate GPA */
 return gpa; /* return the GPA */
}
 int main()
{ struct Student s[] = {{101, "John", 3.33}, {102, "Doe", 3}, {103, "Peter", 3.1}, {EOD,
NULL, EOD}}; /* terminating element */
 struct Student *sp;
 sp = s;
 func1(3, s);
 func1(3, sp);
 func2(3, s);
 func2(3, sp);
 printf("GPA for students in the list is %.2f\n", averageGPA(s));
return 0;
}
```