Lab 8

Task 1: To define a structure called student includes:

- a char array named firstname (length 20 chars)
- a char array named lastname (length 20 chars)
- an integer named age

Paste your code in the box below

```
struct student {
     char firstname[20];
     char lastname[20];
     int  age;
};
```

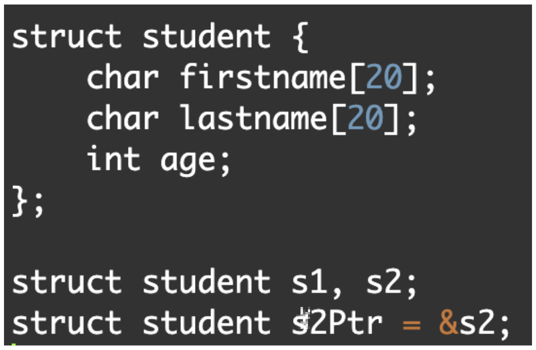
Task 2: With the definition in Task 1, to declare two variables (s1, s2) of the structure, and then to assign listed values:

- s1: Rupert Grint (age 33)
- s2: Emma Watson (age 31)

Paste your code in the box below

```
s1 = (struct student) {"Rupert", "Grint", 33};
s2 = (struct student) {"Emma", "Watson", 31};
```

Task 3: Assume we have definition as:



```
struct student {
    char firstname[20];
    char lastname[20];
    int age;
};

struct student s1, s2;
struct student s2Ptr = &s2;
```

- To assign listed values using s1 and s2Ptr
- [Hint: using assignment (=) and function strcpy]
    - s1: Rupert Grint (age 33)
    - s2: Emma Watson (age 31)

Paste your code in the box below

```
strcpy(s1.firstname, "Rupert");
strcpy(s1.lastname, "Grint");
s1.age = 33;
```

```
strcpy(s2Ptr->firstname, "Emma");
strcpy(s2Ptr->lastname, "Watson");
s2Ptr->age = 31;
```

Task 4: To define a structure with one integer, one char array of length 3 and one double, make a guess of its length, and then to print its length (using sizeof) to see if your guess is correct.
[Hint: Usually, an int takes 4 bytes, a char array of length 3 takes 3 bytes, and a double takes 8 bytes]
Paste your code in the box below

```
struct ure {
     int eger;          // 4
     char acter[3];     // 3
     double precision; // 8
}; // 15 rounds up to 16

printf("%zu\n", sizeof(struct ure));
```

Your guess of the length of the structure is

```
16
```

The output of the length of the structure is

```
16
```

Task 5: To define a structure of your choice including a pointer to itself, and then to define a name for it (using typedef)

```
typedef struct EmptyNode EmptyNode;
struct EmptyNode {
     EmptyNode* next;
};
```

Task 6: To define a union called student_info includes:

- a char array named name (length 20 chars)
- a char named gender
- an integer named age

Paste your code in the box below

```
union student_info {
     char name[20];
     char gender;
     int  age;
};
```

And then to define a struct named student includes three instances of the union, one for each purpose.

Paste your code in the box below.

```
struct student {
     union student_info field_1;
     union student_info field_2;
     union student_info field_3;
};
```

Task 7: With the struct defined in your Task 6, to declare one variable of the struct and to give values as below:

name: Jenny Simpson ('F' for female) age: 26

Paste your code in the box below

```
struct student jenny;
strcpy(jenny.field_1.name, "Jenny Simpson");
jenny.field_2.gender = 'F';
jenny.field_2.age = 26;
```

Task 8: define a **program** declares an int n with initial value 7, and then prints all of listed values with "%d" without changing the value in n:

(1) Bitwise AND for n and 11 (eleven in decimal)
(2) Bitwise OR for n and 0x1001 (1001 in hex)
(3) Bitwise XOR for n and 0x1111 (1111 in hex)

Paste your code in the box below

```
#include <stdio.h>

int main(void) {
     int n = 7;

     printf("%d\n%d\n%d\n",
          n & 11,
          n | 0x1001,
          n ^ 0x1111);

     return 0;
}
```

Task 9: define a **program** declares an int n with initial value 7, and then prints all of listed values with “%d” without changing the value in n:

(1) Left shift value in n for two bits (To print shifted value without changing n)
(2) Right shift value in n for two bits (To print shifted value without changing n)
(3) Bitwise NOT (One’s compliment) of n

Paste your code in the box below

```
#include <stdio.h>

int main(void) {
     int n = 7;

     printf("%d\n%d\n%d\n",
          n << 2,
          n >> 2,
          ~n);

     return 0;
}
```

Task 10: define a **program** declares two enums named seasons and weekdays with values as:

- seasons: SPRING (1), SUMMER(2), FALL (3), WINTER(4)
- weekdays: SUNDAY(0), MONDAY(1), …, SATURDAY(6)

And then, to print the correspond values for Wednesday and Fall

Paste your enum definitions code in the box below

```
#include <stdio.h>

enum seasons {
     SPRING = 1, SUMMER, FALL, WINTER
};

enum weekdays {
     SUNDAY,
     MONDAY,
     TUESDAY,
     WEDNESDAY,
     THURSDAY,
     FRIDAY,
     SATURDAY
};
```

Paste your main function code in the box below

```
int main(void) {
     printf("%d\n", WEDNESDAY);
     printf("%d\n", FALL);

     return 0;
}
```