الجمهورية العربية السورية
الجامعة الافتراضية السورية

الجامعة الافتراضية السورية

وظيفة مقرر مبادئ البرمجة

اشراف الأستاذ وسيم يوسف

إعداد الطالب:

الاسم	الرقم الجامعي	رقم الصف
عمار خفاجة	Ammar_376444	C1

الفهرس

المسألة الاولى: برنامج حساب أهلية الحصول على قرض شخصي 2
المسألة الثانية: برنامج لعبة الجمع الذكي 3
المسألة الثالثة: برنامج ادارة بيانات طلاب الجامعة 4
تابع اضافة طالب جديدadding_process 4
تابع يأخذ وسيطين ويعيد الاسم الثلاثيget_name 5
تابع يأخذ وسيط ويعيد أسم الطالب الذي حصل على أعلى معدل تراكميget_high_average 5
تابع يأخذ وسيط ويعيد قائمة بأسماء الطلاب مرتبة تنازلياً وفقاً لقيمة المعدل التراكميlist_name_descending 6
تابع يأخذ وسيط ويعيد قائمة بأسماء الطلاب الثلاث ذات المعدل الأعلى مرتبة تصاعدياًhigh_three_average 6
تابع يأخذ وسيطين ويقوم بكتابة معلومات القاموس في الملفwrite_to_file 7

المسألة الاولى: برنامج حساب أهلية الحصول على قرض شخصي

خوارزمية الحل باستخدام لغة الخوارزميات Pseudo code

1. Start
2. 0 → age
   0 → monthly_income
   0→ loan_value
   0 →monthly_payment
   0 → insurance
   0 → total_monthly_payment
   [] → ban_reason
3. Read age, monthly_income, loan_value
4. while true

   if age is valid value (numeric and positive value)

   store it

   break

   else:

   read age

5. same steps of number 4 repeated for monthly income + loan value
6. If age < 21 and age >=65

   اضافة " العمر غير مناسب " → ban_reason

7. If monthly_income<3000
   اضافة "الدخل الشهري غير كاف" → ban_reason
8. If loan_value<=5000

   اضافة "قيمة القرض أقل من المسموح" → ban_reason

9. If len (ban_reasons) ==0
   A. loan_value * 0.1 → monthly_payment
   B. If loan_value <20000
      loan_value * 0.015 →insurance
      else
      loan_value * 0.01 →insurance
   C. Insurance + monthly_payment → total_monthly_payment
   D. If total_monthly_payment > (monthly_income * 0.3)

      Write "القسط الشهري يتجاوز 30% من الدخل الشهري"

```
        Else
            Write "أنت مؤهل للحصول على قرض"
                Write " قيمة القسط الشهري: " + monthly_payment
                Write " قيمة قسط التأمين: " + insurance
                Write " المجموع الكلي للقسط الشهري: " + total_monthly_payment
    else
        write ban_reasons
10. End
```

المسألة الثانية: برنامج لعبة الجمع الذكي

خوارزمية الحل باستخدام لغة الخوارزميات Pseudo code

1. Start
2. [] →num_list
   [] →pair_list
   [] → unique_list_value
   0 →target_number
3. For z←0 to 9
   Read value
   Check value is positive number
   Value →target_list[z]
4. Read target_number
5. For x←0 to 8
   For y←x+1 to 9
   If num_list[x]+ num_list[y]==target_number
   [num_list[x], num_list[y]] added to pair_list
6. If length of pair_list==0
   Print("لا يوجد أي زوج يحقق الهدف")

   Else

   For x in pair_list

   If x not in unique_list_value

   Add x → to unique_list_value

Write unique_list_value

7. End

المسألة الثالثة: برنامج ادارة بيانات طلاب الجامعة

I have already declared a dictionary named {container_data} to store student information

تابع اضافة طالب جديد adding_process

1. Start
2. Call adding_process()
3. Call add_student()
    a. Call check_id()
    Read id
    While true
        If id is digit
            If id is positive
                If id is not used before
                    return int(id)
                Else
                    Write("id is used")
        Read id
    b. Id→ID
    c. Call check_name()
        Read name
        While true
            If all chars in name are alpha or spaces
                If len(name)>=3
                    Remove spaces from beginning and end
                    Return name
            Read name
    d. Call check_average()
    Read average
    While true

```
If average is digit
    If average >0 and average <=100
        Return int(average)
Read average
```

4. [name,average] →Container_data[id]
5. Read desire

```
While true
    While desire not in ["Y","y","N","n"]
        Read desire
    If  "y" == desire or "Y" == desire
        Call add_student()
        Read desire
    If  "n" == desire or "N" == desire
        Return
```

6. End

تابع يأخذ وسيطين ويعيد الاسم الثلاثي get_name

1. Start
2. Call get_name(dict,id)
   a. If len(dect)==0

```
return "القاموس فارغ"
```

   b. Try

```
Return dict[id][0]
```

```
Catch
    Return " الرقم الجامعي غير موجود"
```

3. End

تابع يأخذ وسيط ويعيد أسم الطالب الذي حصل على أعلى معدل تراكمي get_high_average

1. Start
2. Call get_high_Average(dict)
   a. If len(dict)==0

```
Return "القاموس فارغ"
```

b. [] →Key_of_high_Average
c. 0 → high_Average
    i. For id in dict.keys()
        If average of id>high_average
            Average of id → high_Average
            Delete all items of key_of_high_Average list
            Add id to key_of_high_value
        Elif average of id == high_average
            Add id to key_of_high_value

    ii. Return names from dict of ids that store in key_of_high_value as a list
3. End

تابع يأخذ وسيط ويعيد قائمة بأسماء الطلاب مرتبة تنازلياً وفقاً لقيمة المعدل التراكمي
list_name_descending

1. Start
2. Call list_name_descending(dict)
    a. If len(dict)==0
        Return "القاموس فارغ"
    b. Sort dict.items() list descending according to average →sorted_list
    c. [] → result
    d. For item in sorted_list
        Add name of item to result
    e. Return result
3. End

تابع يأخذ وسيط ويعيد قائمة بأسماء الطلاب الثلاث ذات المعدل الأعلى مرتبة تصاعدياً
high_three_average

1. Start
2. Call high_three_Average(dict)
    a. If len(dict)==0
        Return "القاموس فارغ"
    b. Sort dict.items() list according to average →sorted_list
    c. Slice sorted_list and take last three items
    d. [] → result
    e. For item in sorted_list

Add name of item to result

f. Return result

3. End

تابع يأخذ وسيطين ويقوم بكتابة معلومات القاموس في الملف write_to_file

4. Start
5. Call write_to_file(dict, file_name)
    a. If len(dict)==0
        Return "القاموس فارغ"
    b. Open file in write mode
    c. Write headers to file
    d. For key in dict.keys()
        a. Write to file the following (key, name of key, average of key)
    e. Write ("data stored in the file)
6. End