Question 1.

```
libname orion '/folders/myfolders/Dataset';
proc sort data=orion.payroll
  out=Work.sorted;
  by dept;
run;

data work.budget;
  set Work.sorted;
  keep Dept PAYROLL;
  if WageCat='S' then YEARLY = WageRate * 12;
  if WageCat='H' then YEARLY = WageRate * 2000;
  by Dept;
  if First.Dept then PAYROLL=0;
  PAYROLL+YEARLY;
  if Last.Dept;
  format PAYROLL DOLLAR12.2;
  rename Dept=Department;
run;

proc format;
  value $deptfmt 'ADM10' = 'ADMN10'
                 'ADM20' = 'ADMN20'
                  'ADM30' = 'ADMN30';
run;

proc print data=work.budget noobs;
  title 'PAYROLL BY DEPARTMENT';
  footnote 'Reported by SAS student T1 2021';
  format Department $deptfmt.;
  sum PAYROLL;
run;
```

Question 2.

```
libname orion '/folders/myfolders/Dataset';    /*assign orion to set orion.Mobile below*/
data compare;
    set orion.Mobile;
    array mth{12} Month_1-Month_12;     /*This is a syntax error as month 1 column does not exist. Change Month 1 to Month_12.*/
    array grp{12} $Grp1-Grp12;
    do i=1 to 12;                        /*As there are 12 groups, change 8 to 12.*/
        if mth{i} < Minutes            /*ERROR: Undeclared array referenced: mth. */
            then grp{i}="Under";            /* add double quotes */
        Else if mth{i} = Minutes    /* ERROR: Undeclared array referenced: mth*/
           then grp{i}="At Plan Limit";      /* add double quotes*/
        Else if mth{i} > Minutes    /* ERROR: Undeclared array referenced: mth*/
          then grp{i}="Over";          /* add double quotes*/
        total=sum(of Month_1-Month_12);  /* ERROR: The funcion AVERAGE is unknown or cannot be accessed*/
        AveMin=total/12;
 /*As the given code does not calculate the average of minutes, rewrite the code to calculate the average of minutes*/
    end;
run;

data subset;
    set compare;
    where Grp1 = 'Over' and Grp2 = 'Over' and Grp3 = 'Over' and Grp4='Over' and Grp5='Over'
               and Grp6='Over' and Grp7='Over' and Grp8='Over' and Grp9='Over' and Grp10='Over'
               and Grp11='Over' and Grp12='Over';
    keep ID Minutes AveMin;
run;

option date;
proc print data=work.subset noobs;
    title 'Mobile Plan Usage';
run;
```

Question 3.

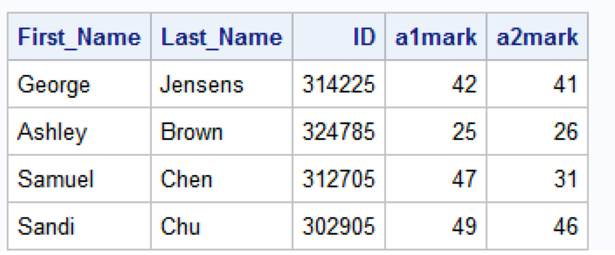
a) Creating a user-defined format to print 1 as ‘Male’ and a 2 as a ‘Female’ is better than creating a new character variable in a DATA step and then printing the character value because it allows us to display the values directly without creating a new data set. If we were to create a new character variable in a DATA step to take on values which had existing data set that were large, this would be a waste of computer resources. Furthermore, user-defined formats using PROC format do not allow one-to-many or many-to-many mappings, preventing data values from taking on more than one label. While PROC format provides a check on the assignment of labels to data values, when using DATA step along with IF-THEN-ELSE to assign labels, SAS log does not indicate if data values are pointing to more than one label. Thus, creating a user-defined format may be better than creating a new character variable in a DATA step and then printing the character value.

b) The advantage of using the CLASS statement rather than the BY statement with PROC MEANS is that the data does not need to be sorted and does not restrict features of PROC MEANS. When using the BY statement, data must be sorted before the PROC MEANS. Furthermore, when BY statements are used, the functionality of PROC MEANS is restricted, for example, the use of way and type statements, due to the limited n-way interaction between the class variables. Thus, the CLASS statement may be preferred with using PROC MEANS rather than the BY statement.

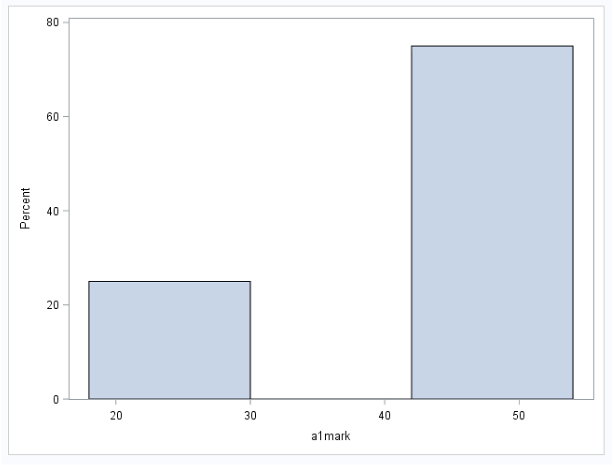
Question 4.

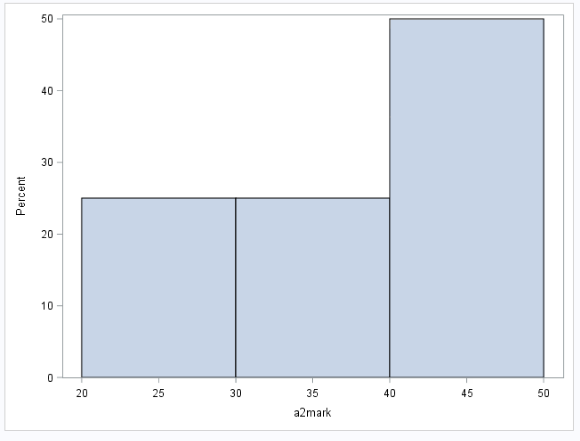
a)

```
data raw_marks1a;
  length First_Name $6 Last_Name $7 ID 6;
  input First_Name $ Last_Name $ID a1 a1_val1 a1_val2 al_val3 al_val4 al_val5
  a2 a2_val1 a2_val2 a2_val3 a2_val4 a2_val5;
  datalines;
George Jensens 314225
a1 7 10 7 10 8
a2 7 7 8 9 10
Ashley Brown 324785
a1 2 1 6 7 9
a2 7 1 8 6 4
Samuel Chen 312705
a1 10 9 9 9 10
a2 8 7 9 7 0
Sandi Chu 302905
a1 10 10 10 10 9
a2 9 9 10 9 9
run;

data marks1a;
  set raw_marks1a;
  a1mark = sum(a1_val1, a1_val2, al_val3, al_val4, al_val5);
  a2mark = sum(a2_val1, a2_val2, a2_val3, a2_val4, a2_val5);
run;

proc print data=marks1a noobs;
  var First_Name Last_Name ID a1mark a2mark;
run;

proc sgplot data=marks1a;
  histogram a1mark;
run;

proc sgplot data=marks1a;
  histogram a2mark;
run;
```



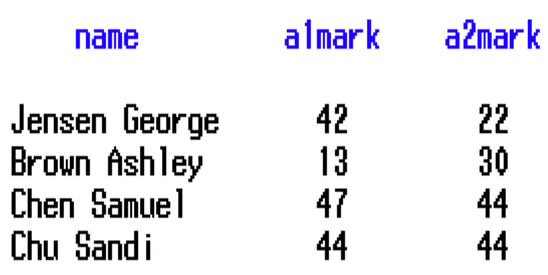
First_Name	Last_Name	ID	a1mark	a2mark
George	Jensens	314225	42	41
Ashley	Brown	324785	25	26
Samuel	Chen	312705	47	31
Sandi	Chu	302905	49	46



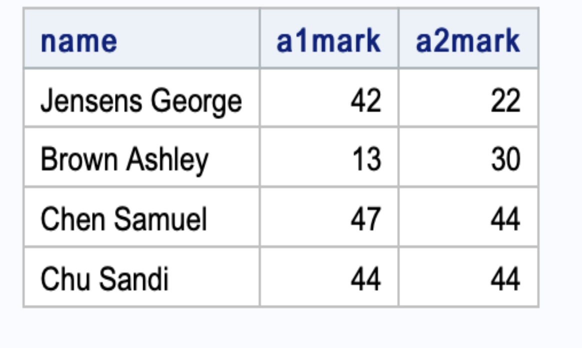


b)

```
data marks1b;
   infile '/folders/myfolders/Dataset/marks1b.csv' dlm=',,:';
   length First_Name $6 Last_Name $7 ID 6;
   input First_Name $ Last_Name $ID a1 a1_val1 a1_val2 al_val3 al_val4 al_val5
         a2 a2_val1 a2_val2 a2_val3 a2_val4 a2_val5;
   a1mark = sum(a1_val1, a1_val2, al_val3, al_val4, al_val5);
   a2mark = sum(a2_val1, a2_val2, a2_val3, a2_val4, a2_val5);
   name = catx(' ', First_Name, Last_Name);
run;

proc print data=work.marks1b noobs;
   var name a1mark a2mark;
run;
```



```
    name            a1mark    a2mark

Jensen George         42        22
Brown Ashley          13        30
Chen Samuel           47        44
Chu Sandi             44        44
```



name	a1mark	a2mark
Jensens George	42	22
Brown Ashley	13	30
Chen Samuel	47	44
Chu Sandi	44	44