In [2]:

```
# python program to find the average of odd numbers in the given list
data=[68, 71, 21, 68, 32, 80, 64, 52, 92, 97, 93, 37, 61, 62, 23, 81, 67, 84, 33, 41, 92, 5
43, 80, 74, 69, 92, 79, 78, 70, 31, 34, 79, 73, 77, 96, 87, 53, 47, 85, 58, 58, 87, 44, 29,
71, 57, 96, 74, 30, 65, 62, 85, 34, 21, 81, 79, 84, 87, 22, 45, 59, 91, 26, 22, 46, 71, 60,
38, 82, 95, 77, 55, 37,61, 59, 89, 61, 75, 97, 72, 58, 23, 78, 20, 95, 67, 59, 56, 93, 82,3
95, 90, 80, 78, 73, 91, 52, 59, 96, 46, 94, 99,78, 53, 40, 67, 76, 73, 46, 77, 40, 34, 38,
61, 76,21, 90, 52, 65, 36, 93, 55, 86, 95, 65, 30, 89, 86, 20, 92, 46, 58,75, 43, 57, 53, 4
58, 31, 73, 71, 63, 70, 72, 78,26, 39, 35, 58, 84, 56, 24, 73, 36, 75, 32, 35, 56, 87, 87,
38, 94, 36, 90, 70, 65, 34, 61, 44, 85, 41, 50, 23, 34,20, 87, 84, 30, 21, 84, 65, 85, 88,
41, 35, 71]
sm,cnt=0,0

for x in data:
    if x%2!=0:
        sm+=x
        cnt+=1

print(" avg of all odd numbers is ",str(sm/cnt))
```

```
 avg of all odd numbers is  61.97391304347826
```

In [3]:

```
#python program to find the third largest number in the given list
import sys
def thirdLargest(arr, arr_size):

    # There should be
    # atleast three elements
    if (arr_size < 3):

        print(" Invalid Input ")
        return


    # Find first
    # largest element
    first = arr[0]
    for i in range(1, arr_size):
        if (arr[i] > first):
            first = arr[i]

    # Find second
    # largest element
    second = -sys.maxsize
    for i in range(0, arr_size):
        if (arr[i] > second and
            arr[i] < first):
            second = arr[i]

    # Find third
    # largest element
    third = -sys.maxsize
    for i in range(0, arr_size):
        if (arr[i] > third and
            arr[i] < second):
            third = arr[i]

    print("The Third Largest",
        "element is", third)

# Driver Code
arr = [68, 71, 21, 68, 32, 80, 64, 52, 92, 97, 93, 37, 61, 62, 23, 81, 67, 84, 33, 41, 92,
43, 80, 74, 69, 92, 79, 78, 70, 31, 34, 79, 73, 77, 96, 87, 53, 47, 85, 58, 58, 87, 44, 29,
71, 57, 96, 74, 30, 65, 62, 85, 34, 21, 81, 79, 84, 87, 22, 45, 59, 91, 26, 22, 46, 71, 60,
38, 82, 95, 77, 55, 37,61, 59, 89, 61, 75, 97, 72, 58, 23, 78, 20, 95, 67, 59, 56, 93, 82,3
95, 90, 80, 78, 73, 91, 52, 59, 96, 46, 94, 99,78, 53, 40, 67, 76, 73, 46, 77, 40, 34, 38,
61, 76,21, 90, 52, 65, 36, 93, 55, 86, 95, 65, 30, 89, 86, 20, 92, 46, 58,75, 43, 57, 53, 4
58, 31, 73, 71, 63, 70, 72, 78,26, 39, 35, 58, 84, 56, 24, 73, 36, 75, 32, 35, 56, 87, 87,
38, 94, 36, 90, 70, 65, 34, 61, 44, 85, 41, 50, 23, 34,20, 87, 84, 30, 21, 84, 65, 85, 88,
41, 35, 71]
n = len(arr)
thirdLargest(arr, n)
```

```
The Third Largest element is 96
```

In [7]:

```
# python program to find the count of even and odd numbers in the given list
data=[68, 71, 21, 68, 32, 80, 64, 52, 92, 97, 93, 37, 61, 62, 23, 81, 67, 84, 33, 41, 92, 5
43, 80, 74, 69, 92, 79, 78, 70, 31, 34, 79, 73, 77, 96, 87, 53, 47, 85, 58, 58, 87, 44, 29,
71, 57, 96, 74, 30, 65, 62, 85, 34, 21, 81, 79, 84, 87, 22, 45, 59, 91, 26, 22, 46, 71, 60,
38, 82, 95, 77, 55, 37,61, 59, 89, 61, 75, 97, 72, 58, 23, 78, 20, 95, 67, 59, 56, 93, 82,3
95, 90, 80, 78, 73, 91, 52, 59, 96, 46, 94, 99,78, 53, 40, 67, 76, 73, 46, 77, 40, 34, 38,
61, 76,21, 90, 52, 65, 36, 93, 55, 86, 95, 65, 30, 89, 86, 20, 92, 46, 58,75, 43, 57, 53, 4
58, 31, 73, 71, 63, 70, 72, 78,26, 39, 35, 58, 84, 56, 24, 73, 36, 75, 32, 35, 56, 87, 87,
38, 94, 36, 90, 70, 65, 34, 61, 44, 85, 41, 50, 23, 34,20, 87, 84, 30, 21, 84, 65, 85, 88,
41, 35, 71]
even_count, odd_count = 0, 0

for num in data:

    if num % 2==0:
        even_count +=1
    else:
        odd_count +=1

print("Even numbers in the list: ", even_count)
print("Odd numbers in the list: ",odd_count)
```

```
Even numbers in the list:  105
Odd numbers in the list:  115
```

In [20]:

```
#Final assessment data analysis TEST
import pandas as pd
import numpy as np
```

In [9]:

```
df_test=pd.read_csv('Test.csv')
```

In [12]:

```
df_test.head()
```

Out[12]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_I
0	FDW58	20.750	Low Fat	0.007565	Snack Foods	107.8622	
1	FDW14	8.300	reg	0.038428	Dairy	87.3198	
2	NCN55	14.600	Low Fat	0.099575	Others	241.7538	
3	FDQ58	7.315	Low Fat	0.015388	Snack Foods	155.0340	
4	FDY38	NaN	Regular	0.118599	Dairy	234.2300	

In [14]:

```
df_test.tail()
```

Out[14]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outl
5676	FDB58	10.5	Regular	0.013496	Snack Foods	141.3154	
5677	FDD47	7.6	Regular	0.142991	Starchy Foods	169.1448	
5678	NCO17	10.0	Low Fat	0.073529	Health and Hygiene	118.7440	
5679	FDJ26	15.3	Regular	0.000000	Canned	214.6218	
5680	FDU37	9.5	Regular	0.104720	Canned	79.7960	

In [27]:

```
MRP = df_test['Item_MRP']
name = df_test['Item_Identifier']
weight = df_test['Item_Weight']
fat = df_test['Item_Fat_Content']
visibility = df_test['Item_Visibility']
Type = df_test['Item_Type']
identifier = df_test['Outlet_Identifier']
year = df_test['Outlet_Establishment_Year']
size = df_test['Outlet_Size']
L_Type = df_test['Outlet_Location_Type']
Outlet_Type = df_test['Outlet_Type']
```

In [31]:

```
avg_col = np.mean(MRP,0)
print(avg_col)
```

```
141.02327340256954
```

In [32]:

```
data_test = [name, Type, MRP, print('average of the MRP =',avg_col)]
df_test.head(10)
```

```
average of the MRP = 141.02327340256954
```

Out[32]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_I
0	FDW58	20.750	Low Fat	0.007565	Snack Foods	107.8622	
1	FDW14	8.300	reg	0.038428	Dairy	87.3198	
2	NCN55	14.600	Low Fat	0.099575	Others	241.7538	
3	FDQ58	7.315	Low Fat	0.015388	Snack Foods	155.0340	
4	FDY38	NaN	Regular	0.118599	Dairy	234.2300	
5	FDH56	9.800	Regular	0.063817	Fruits and Vegetables	117.1492	
6	FDL48	19.350	Regular	0.082602	Baking Goods	50.1034	
7	FDC48	NaN	Low Fat	0.015782	Baking Goods	81.0592	
8	FDN33	6.305	Regular	0.123365	Snack Foods	95.7436	
9	FDA36	5.985	Low Fat	0.005698	Baking Goods	186.8924	

In [33]:

```
#Final assessment data analysis TRAIN
import pandas as pd
```

In [34]:

```
df_train = pd.read_csv('Train.csv')
```

In [35]:

```
df_train
```

Out[35]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outl
0	FDA15	9.300	Low Fat	0.016047	Dairy	249.8092	
1	DRC01	5.920	Regular	0.019278	Soft Drinks	48.2692	
2	FDN15	17.500	Low Fat	0.016760	Meat	141.6180	
3	FDX07	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	
4	NCD19	8.930	Low Fat	0.000000	Household	53.8614	
...	...	...	...	...	...	...	
8518	FDF22	6.865	Low Fat	0.056783	Snack Foods	214.5218	
8519	FDS36	8.380	Regular	0.046982	Baking Goods	108.1570	
8520	NCJ29	10.600	Low Fat	0.035186	Health and Hygiene	85.1224	
8521	FDN46	7.210	Regular	0.145221	Snack Foods	103.1332	
8522	DRG01	14.800	Low Fat	0.044878	Soft Drinks	75.4670	

8523 rows × 12 columns

In [36]:

```
len(df_train['Item_Type'])
df_train['Item_Type'].count()
```

Out[36]:

```
8523
```

In [37]:

```
#value count of Item_Type
df_train["Item_Type"].value_counts()
```

Out[37]:

```
Fruits and Vegetables    1232
Snack Foods              1200
Household                 910
Frozen Foods              856
Dairy                     682
Canned                    649
Baking Goods              648
Health and Hygiene        520
Soft Drinks               445
Meat                      425
Breads                    251
Hard Drinks               214
Others                    169
Starchy Foods             148
Breakfast                 110
Seafood                    64
Name: Item_Type, dtype: int64
```

In [38]:

```
#percentage of Item_Type
df_train['Item_Type'].value_counts(normalize=True).round(3)
```

Out[38]:

```
Fruits and Vegetables    0.145
Snack Foods              0.141
Household                0.107
Frozen Foods             0.100
Dairy                    0.080
Canned                   0.076
Baking Goods             0.076
Health and Hygiene       0.061
Soft Drinks              0.052
Meat                     0.050
Breads                   0.029
Hard Drinks              0.025
Others                   0.020
Starchy Foods            0.017
Breakfast                0.013
Seafood                  0.008
Name: Item_Type, dtype: float64
```

In [39]:

```
df_train[df_train.index.isin([222])]
```

Out[39]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outle
222	FDH28	15.85	Regular	0.110031	Frozen Foods	37.2506	

In [55]:

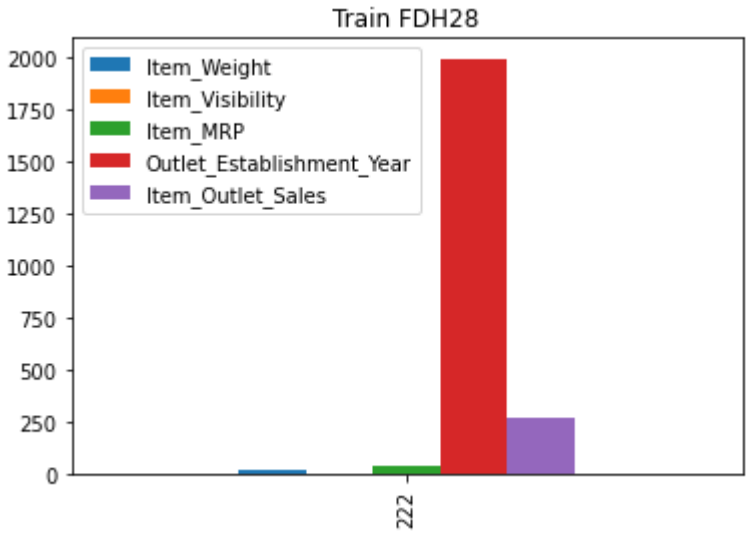
```
df_train_222 = df_train[df_train.index.isin([222])]
```

In [64]:

```
df_train_222.plot(kind='bar', title='Train FDH28')
```

Out[64]:

```
<AxesSubplot:title={'center':'Train FDH28'}>
```



In [41]:

```
df_train_bartype = df_train[df_train.index.isin([723, 582, 392, 7000])]
```

In [44]:

```
df_train_bartype.plot(kind='bar')
```

Out[44]:

<AxesSubplot:>