Program :-

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import LabelEncoder
from sklearn.decomposition import PCA

data = pd.read_csv("/home/student/Desktop/sinan/heart.csv")
label_encoder = LabelEncoder()
for col in data.columns:
    data[col] = label_encoder.fit_transform(data[col])

X = data.drop("HeartDisease", axis=1)

y = data["HeartDisease"]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("\n\nLinear where Constants = 1")
svm_model_linear = SVC(kernel='linear', C=1)
svm_model_linear.fit(X_train, y_train)

y_pred = svm_model_linear.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)


print("\n\nLinear where Constants = 100")
svm_model_linear = SVC(kernel='linear', C=100)
svm_model_linear.fit(X_train, y_train)

y_pred = svm_model_linear.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)


print("\n\RBF where Constants = 2")
svm_model_rbf = SVC(kernel='rbf', C=2)
svm_model_rbf.fit(X_train, y_train)

y_pred = svm_model_rbf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
class_report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)


print("\n\RBF where Constants = 6")
svm_model_rbf = SVC(kernel='rbf', C=6)
svm_model_rbf.fit(X_train, y_train)

y_pred = svm_model_rbf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

Output :-

```
(base) student@cseadmin:~/Desktop/sinan$ python3 svm.py


Linear where Constants = 1
Accuracy: 0.8260869565217391
Confusion Matrix:
 [[67 10]
 [22 85]]
Classification Report:
             precision   recall  f1-score  support

         0      0.75     0.87     0.81       77
         1      0.89     0.79     0.84      107

   accuracy                       0.83      184
  macro avg      0.82     0.83     0.82      184
weighted avg      0.84     0.83     0.83      184



Linear where Constants = 100
Accuracy: 0.8260869565217391
Confusion Matrix:
 [[67 10]
 [22 85]]
Classification Report:
             precision   recall  f1-score  support

         0      0.75     0.87     0.81       77
         1      0.89     0.79     0.84      107

   accuracy                       0.83      184
  macro avg      0.82     0.83     0.82      184
```

```
weighted avg       0.84      0.83      0.83       184


\RBF where Constants = 2
Accuracy: 0.7391304347826086
Confusion Matrix:
 [[58 19]
 [29 78]]
Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.75      0.71        77
           1       0.80      0.73      0.76       107

    accuracy                           0.74       184
   macro avg       0.74      0.74      0.74       184
weighted avg       0.75      0.74      0.74       184


\RBF where Constants = 6
Accuracy: 0.7771739130434783
Confusion Matrix:
 [[61 16]
 [25 82]]
Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.79      0.75        77
           1       0.84      0.77      0.80       107

    accuracy                           0.78       184
   macro avg       0.77      0.78      0.77       184
weighted avg       0.78      0.78      0.78       184


(base) student@cseadmin:~/Desktop/sinan$
```