Rapport de Projet compile

1. **Les fonction :**
   1. **Partie Symantique :**
      1. **Traitement de la table de symbole :**
2. **void init();**

Pour initialiser la table de symbole.

1. **int hachage(char chaine[]);**

Pour hacher les variables stocker dans la table de symbole en codant chaque syllabe d’identifiant.

1. **int rechercher(char nom [], element \*\* in);**

Pour la recherche des varibale dans la table desymbole.

1. **void inserer (char nom[], int code,char nature[30],char taille[255]);**

Pour l’insertion des variables avec ses différentes informations dans la table de symbole.

1. **int gettype(char nom[]);**

Pour retourner le type de la variable.

1. **char\* substring(char string[]);**

Pour extraire une sous-chaine.

1. **void afficher();**

Pour afficher le contenue de la table de symbole.

1. **void used(char nom[]);**

Pour signé l’utilisation de la variable.

1. **void delete\_unused();**

Pour supprimer tous les variables non utilisées.

1. **void modifier( char nom[], int code);**

Pour changer le type d’une variable en utilisant un code.

1. **char\* getValeur(char\* idf);**

Pour retourner la valeur d’une variable.

1. **void setValeur(char\* idf,char\* valeur);**

Pour la mise à jour de la valeur d’une variable.

1. **char\* typeVar(int i);**

Pour retourner le type correspondant d’après le code numérique.

1. **void err\_taille\_tab(int type,char val[]);**

Pour informer le dépassement de taille des variables de type tableau.

1. **void err\_incompa\_typ\_tab(int typtab,int typvar);**

Pour informer l’incompatibilité des variables de type tableau.

1. **void err\_incompa\_typ\_var(int typdec,int typvar);**

Pour informer l’incompatibilité des variables.

1. **int declared(char nom[]) ;**

Pour informer si la variable est déclarer ou non.

1. **int max(int a, int b) ;**

Pour désigner le max entre les deux variables.

* + 1. **Traitement des quadruplés** **:**

1. **void generer(char\* a, char\* b, char\* c, char\* d);**

Pour insérer les informations dans le quadruplés.

1. **void quadL(int i, char\*b, char\* c, char\* d);**

Pour générer les quadruplés **de comparaison d’égalité et différence** utiliser dans les conditions de :

* **IF**
* **WHILE**
* **FOR**

1. **char\* convert(int i);**

Pour convertir l’entier en chaine de caractère.

1. **void quadC(int i, char\* b, char\* c, char\* d);**

Pour générer les quadruplés **de comparaison de supériorité et d’infériorité** utiliser dans les conditions de :

* **IF**
* **WHILE**
* **FOR**

1. **void afficher\_qdr();**

Pour afficher l’intégralité des quadrulés.

1. **void delete\_quad();**

Pour supprimer les quadruplés des variable et variable temporaire non utilisées .

* + 1. **Traitement de code assembleur :**

1. **void assembler();**
2. **void propaDeCopie();**
3. **void propaDeCopie();**
4. **void simplification\_algebrique();**
5. **void elimination\_code\_inutile();**
6. **void Maj(int nQc, int jmp);**